

AD-A080 371

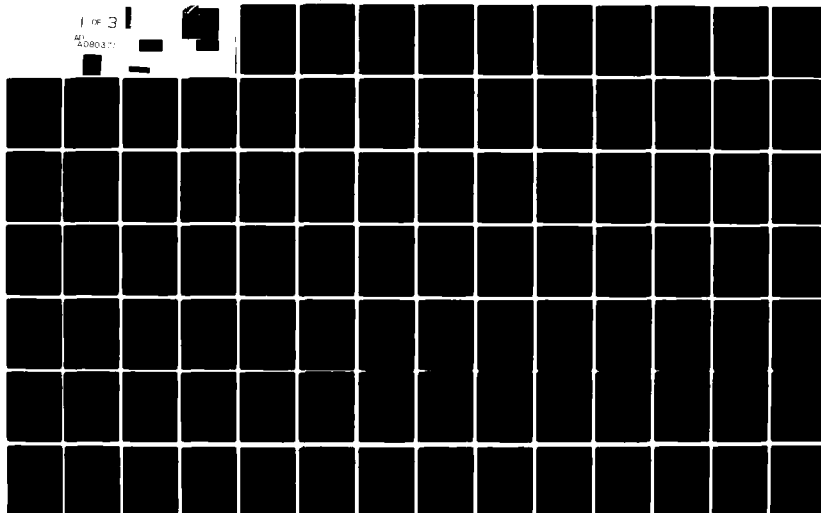
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/4
MODEL ORDER REDUCTION USING THE BALANCED STATE REPRESENTATION! --ETC(U)
DEC 79 J R MCCLENDON
AFIT/6E/EE/79-22

UNCLASSIFIED

NL

1 OF 3

AD-A080371





DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DDC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

AFIT/GE/EE/79-22

(6) MODEL ORDER REDUCTION USING THE
BALANCED STATE REPRESENTATION:
THEORY, APPLICATION, AND INTER-
ACTIVE SOFTWARE IMPLEMENTATION.

(9) Master's THESIS, (Robert)

(14) AFIT/GE/EE/79-22 (James McClendon)

(10) Lieutenant USAF

(11) Dec 79

(12) 214

RECEIVED
A

Approved for public release; distribution unlimited

012225

mt

AFIT/GE/EE/79-22

MODEL ORDER REDUCTION USING THE
BALANCED STATE REPRESENTATION: THEORY
APPLICATION, AND INTERACTIVE SOFTWARE IMPLEMENTATION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

By

James R. McClendon, B.S.
Lieutenant USAF

Graduate Guidance and Control
December 1979

AFIT/GE/EE/79-22	
Author	James R. McClendon
Title	MODEL ORDER REDUCTION USING THE BALANCED STATE REPRESENTATION: THEORY APPLICATION, AND INTERACTIVE SOFTWARE IMPLEMENTATION
Branch	AFIT/GE/EE
Dist	A
Special	23

Approved for public release; distribution unlimited.

PREFACE

Many model order reduction techniques exist. However, most of these techniques, though based on sound theory, require many man hours of analysis and prior system insight to perform. This thesis implements a model order reduction technique developed by Dr Bruce Moore of the University of Toronto and applies the resulting software package to find a reduced order model of a B-52E flutter control problem which is currently of concern to the Air Force Flight Dynamics Laboratory. This particular technique utilizes a well defined, highly efficient algorithm resulting in impressive reduced order models. Although other techniques may be found providing a reduced model of similar order, this technique has a distinct advantage in that it is highly systematic and may be totally computer automated through the use of "high quality" software routines of IMSL, Eispack or Linpack (Ref 13).

The model order reduction technique currently in use by the Air Force Flight Dynamics Laboratory, Flight Control Division under Dr Bob Schwanz, requires human decisions to determine which modes of a system to retain. The technique to be investigated herein requires no such human decision.

An interactive computer program employing Dr Moore's algorithm as well as offering other related options is presented in

the hope the user may obtain reduced order models representing his full order system and investigate the reduced order model's robustness properties in an efficient manner.

I wish to thank Lt Stanley J. Larimer for his help in interfacing with TOTAL to obtain the time and frequency domain responses that were crucial for evaluation of this model order reduction technique. I would like to thank Capt Jerry Stinson for his help and assistance in learning how to use the ASD Computer System.

My sincere gratitude is extended to Major J. Gary Reid, without whose extensive knowledge and genuine zeal for this project would have made it an impossibility. I would like to also thank Dr Bob Schwanz of the Air Force Flight Dynamics Laboratory for his help and sponsorship of this project.

I wish to thank my wife, Kathy, and my son, Jake, for their understanding and encouragement when I really needed it. Finally, a note of appreciation is necessary for my typist, Eve Vaught, for her outstanding work in typing this thesis.

James R. McClendon

TABLE OF CONTENTS

	Page
PREFACE	ii
LIST OF FIGURES	vi
LIST OF TABLES	viii
LIST OF SYMBOLS	ix
ABSTRACT	x
 I. INTRODUCTION	 1
Background and Problem Statement.	3
Statement of Purpose	5
Scope and Approach	5
Organization of Thesis	6
 II. PRESENTATION OF THE ALGORITHM	 8
Introduction	8
Problem	9
Kalman Theory of Minimal Realization	10
Presentation of the Moore Algorithm	11
The H ₁ Matrix	18
Example	20
Summary of the Moore Algorithm and Internally Balanced Coordinate System.	23
 III. DEVELOPMENT OF INTERACTIVE SOFTWARE	 26
Introduction	26
Program Structure	26
Presentation of Options	29
 IV. APPLICATION OF MOORE ALGORITHM TO SISO EXAMPLE	 44
Introduction	44
Description of Other Algorithms	44
SISO Example	47

TABLE OF CONTENTS (continued)

	Page
V. APPLICATION OF MOORE'S ALGORITHM TO B-52E FLUTTER CONTROL PROBLEM	58
Introduction	58
Original System Model Acquisition	59
Application of Moore "Impulse Balancing" Algorithm	65
Application of Moore "Step Balancing" Algorithm	79
Summary	89
VI. RESULTS AND RECOMMENDATIONS	95
Summary of Results	95
Recommendations for Future Study	99
BIBLIOGRAPHY	101
APPENDIX A: DEFINITION OF THE SINGULAR VALUE DECOMPOSITION	A-i
APPENDIX B: DESCRIPTION OF THE ALGORITHM	B-i
APPENDIX C: MIMO USER'S MANUAL	C-i
APPENDIX D: FORTRAN LISTING OF MIMO	D-i
VITA	200

LIST OF FIGURES

	Page
1. Flow Chart for Impulse Balancing Algorithm.	15
2. Controllability Matrix Max and Min Singular Values and Condition Number vs. Sample Time	32
3. Observability Matrix Max and Min Singular Values and Condition Number vs. Sample Time	34
4. Hankel Matrix Max and Min Singular Values and Condition Number vs. Sample Time	35
5. Unit Impulse Responses for SISO Example	50
6. Impulse Responses for SISO Example	51
7. Magnitude Responses for SISO Example	52
8. Magnitude Responses for SISO Example	54
9. Phase Responses for SISO Example	55
10. Phase Responses for SISO Example	56
11. Sensor Locations	60
12. B-52E CCV Flight Control Surfaces	61
13. Response to .1 Radian Step in Outboard Aileron	66
14. Response to .1 Radian Step in Outboard Aileron	67
15. AW925/OBA To .1 Radian Step Input Using Impulse Balancing	69
16. AW565/OBA To .1 Radian Step Input Using Impulse Balancing	70
17. AW925/OBA Frequency Response Using Impulse Balancing .	71
18. AW925/OBA Phase Response Using Impulse Balancing	72
19. AW565/OBA Frequency Response Using Impulse Balancing .	73

LIST OF FIGURES (continued)

	Page
20. AW565/OBA Phase Response Using Impulse Balancing . . .	74
21. AW925/OBA to .1 Radian Step Input Using Impulse Balancing	76
22. AW565/OBA to .1 Radian Step Input Using Impulse Balancing	77
23. AW925/OBA to .1 Radian Step Input Using Step Balancing .	80
24. AW565/OBA to .1 Radian Step Input Using Step Balancing .	82
25. Frequency Responses (AW925/OBA) Using Step Balancing	84
26. Frequency Responses (AW565/OBA) Using Step Balancing	85
27. Phase Response (AW925/OBA) Using Step Balancing . . .	86
28. Phase Response (AW565/OBA) Using Step Balancing . . .	87
29. AW925/OBA To .1 Radian Step Input Using Step Balancing.	90
30. Frequency Response (AW925/OBA) Using Step Balancing .	91
31. Phase Response (AW925/OBA) Using Step Balancing	92

LIST OF TABLES

	Page
I. Test Case Controllability and Observability Grammians. .	22
II. Comparison of LINPACK Estimate of Condition Number vs. $\sigma_{\max}/\sigma_{\min}$ for a Hilbert Matrix	37
III. Reduced Order Model Transfer Functions for SISO Case	49
IV. The Eigenvalues for the B-52E Test Case	64
V. Singular Values of the H_{INF} Matrix for Impulse Balancing	78
VI. Per-Cent Differences Between 24th, 10th, and 9th Order Systems	83
VII. Singular Values of the H_{INF} Matrix for Step Balancing . .	88

LIST OF SYMBOLS

Symbol	Meaning
$\sigma(i)$	The i th singular value.
A^T	The matrix A transposed.
A^H	The conjugate transpose of A .
$K(A)$	The condition number with respect to inversion of A .
$\ \cdot\ $	Euclidean norm for a vector Subordinate spectral norm for a matrix.
A^+	Generalized inverse of A .
A^{-1}	The inverse of A (square).
$S^\perp, S \in R^n$	Orthogonal complement in R^n .

ABSTRACT

This report investigates a model order reduction technique developed by Dr Bruce Moore of the University of Toronto, as applied to the B-52E flutter control problem currently under study by the Air Force Flight Dynamics Laboratory. The algorithm, which is based upon singular value analysis, is applied to the full twenty-fourth order model yielding an internally balanced representation which is balanced with respect to controllability and observability properties. The system is reduced in order, and comparisons are made between the Moore algorithm model and that obtained via a method used by the Air Force Flight Dynamics Laboratory.

In addition to this investigation, an interactive program is presented which contains the model order reduction algorithm. Other capabilities include: estimation of the condition number with respect to inversion, singular value and condition number plotting vs. sample time for discrete time controllability, observability, and Hankel matrices, frequency response generation, and various special coordinate system transformations.

I. INTRODUCTION

High speed digital computers allow us to investigate systems faster than ever before possible. However, these computers have memory and central processor utilization time restrictions. Many systems such as chemical processes and mechanical structures have large dimension, complex system models. Some systems, such as temperature control systems, are described by partial differential equations. Such systems are infinite dimensional and are called distributed parameter systems. Another example of a distributed parameter system is the flexible wing on an airplane. These partial differential equations must be transformed to ordinary differential equations to yield a finite system model. Often these system models exceed the computer's memory restrictions and a lower order model must be found. For a typical distributed parameter system, system orders in the range of 100 or more are common and are considered large dimension systems.

Two applications for reduced order models are simulation and on-line control. In simulation, the desire is possibly to test equipment or algorithms with the system model. To be feasible in terms of cost and turnaround time, it is highly desirable to utilize a reduced order model which replicates the actual system accurately while being of sufficiently small dimension so as to allow cost effective simulation.

The other application is on-line control. This occurs when a controller is trying to control an actual system on-line in real-time. The model used for control calculations must be small enough so as not to exceed the computation time required for effective control. Often in on-line control, a Kalman filter is required to estimate the states of the system. A reduced order model can be highly beneficial to a Kalman filter design also.

The hardest task in designing a Kalman filter is obtaining an accurate representation of the system (i.e., the model). Unfortunately, the system order is often too great for efficient Kalman filter implementation. For an n -dimensional system, where n is large, the number of multiplies that must be accomplished by a Kalman filter is proportional to n^3 (Ref 22). The Kalman filter gain requires the solving of $n(n+1)/2$ simultaneous equations. Clearly, if the model's order could be reduced while still yielding "good" results, computational burden could be decreased substantially. As an example, if the original system is twenty-fourth order (24 state variables), then 300 simultaneous equations must be solved to calculate the Kalman filter gain. If the model could be reduced to tenth order, only 55 simultaneous equations need be solved.

Other applications such as controller design and actual on-line controller implementation require models of sufficiently low order.

Therefore, the need exists to efficiently and accurately reduce the order of a system model without a prohibitive loss of accuracy.

Background and Problem Statement

Many model order reduction techniques exist. Rogers and Sworder (1971) chose model parameters to optimize a certain cost functional. Hutton and Friedland (Ref 17) suggest the use of Routh approximants. A Routh approximant is defined by making the Routh table coefficients for the approximant match those of the original system to a given order. Chien and Shieh (1968) propose the use of Pade' approximants. A Pade' approximant is defined by choosing the coefficients such that the Taylor series expansions of the approximant and the original system agree in as many terms as possible for a pre-determined system order lower than that of the original system. Many "dominant mode retention" procedures including that of Chidambara (1967) and Marshall (1966) exist. Norm minimization techniques such as the one proposed by El-Attar and Vidyasagar (Ref 14) can be found in the literature. The list is long. The main point, however, is that many techniques exist.

Some of these techniques result in unstable reduced order models when given a stable full-order model. Some are not easily implemented on a computer, and thus many man hours must be spent in deriving the reduced order model. Some techniques work with state space models, others with transfer functions, and still

others deal with the full-order differential equations of the full-order system.

The problem to be investigated in this paper is to implement and analyze a model order reduction algorithm developed by Dr Bruce Moore of the University of Toronto (Ref 24, 25). This algorithm minimizes many of the aforementioned problems associated with model order reduction techniques. The Moore algorithm actually implements working subspaces for the Kalman minimal realization decomposition (Ref 19).

The Moore algorithm actually contains four paths that may be followed, all of which yield reduced order models. These paths include:

- 1) A reduced order model that is "optimal" in response to an impulse input (optimal in the sense that this particular model more closely resembles the original system if both are subjected to an impulse input).

- 2) A reduced order model that is "optimal" in response to a step input.

- 3) A reduced order model which weights the steady state portion of the response to a given input more heavily than the transient portion.

4) A reduced order model which weights the transient portion of the response to a given input more heavily than the steady state portion.

Statement of Purpose

The purpose of this thesis is twofold. The first objective is to develop software, both interactive and batch, which implements the Moore algorithm and investigates its properties in both the frequency and time domains.

The second objective is to take this software and apply it to a practical Air Force Flight Dynamics Laboratory (AFFDL) problem, specifically that of finding an accurate reduced order model for the B-52E flutter control problem.

Scope and Approach

The objectives are accomplished in the following manner. First, batch software is developed to obtain the B-52E flutter control problem's state space model. Next, an interactive program called MIMO is developed to investigate the application of the Moore algorithm to the full order system model. The actual application of the Moore algorithm to the full order system will be accomplished by a batch program because the full order system is twenty-fourth order which exceeds MIMO's tenth order capability. However, the actual coding of the algorithm is identical.

In addition to the application of the Moore algorithm, MIMO provides options for singular value vs. sample time plots (see Section III), discretize the continuous time linear, time-invariant state space system, estimate the condition number with respect to inversion for a square matrix, obtain the discrete-time controllability, observability, and Hankel matrices, obtain the steady state controllability and observability grammians, obtain the frequency response for the balanced system, list the singular values and singular vectors of a matrix as well as obtain various special state coordinate systems (see Section III).

MIMO utilizes overlays which allow a modular design and save computer memory. MIMO is designed to allow the implementation of future options which utilize the data base provided by the program.

Organization of Thesis

The thesis is organized in the following manner:

- (1) Background and algorithm development is presented in Section II.
- (2) In Section III, the interactive program MIMO is explained and its options presented.
- (3) Section IV compares several model order reduction techniques with the Moore algorithm for a third order SISO (single-input-single-output) system.

(4) In Section V, the application of the Moore algorithm to the B-52E flutter control problem is presented. At this point, comparisons are made between the Moore technique and the modified Schwendler and MacNeal technique (Ref 32) currently in use by AFFDL.

(5) Finally, the conclusions and recommendations (Section VI), concerning Moore's algorithm and its applicability to problems of finding reduced order models for the B-52E flutter control problem, are presented.

II. PRESENTATION OF THE ALGORITHM

Introduction

With the advent of digital computers, the ability to analyze large dimension, linear systems has been established. Unfortunately, even with the speed of digital computers, many linear systems possess such a large dimension that real-time computations are prohibited. The model order reduction problem is borne from this type of situation. Though many techniques exist, there is no widely accepted method that completely solves the model order reduction problem. Therefore, the need for an efficient algorithm persists.

The Moore technique takes a novel approach in that it addresses whether a state is close to being redundant as versus merely accepting a given state to be redundant or non-redundant. Transforming a system into a state coordinate system in which this state redundancy is clearly presented is a topic introduced by Kalman (Ref 19) in 1963. The Kalman theory provides a basis for the Moore approach.

The remainder of this section will be organized in the following manner. First, the basic model order reduction problem will be defined. Second, the Kalman theory of minimal realization will be discussed. Next, the relationships between the Kalman theory and the Moore algorithm will be presented. The Moore algorithm actually consists of four paths. Each path will be discussed in detail as well

as when a particular path might be desirable will be presented. Many existing model order reduction procedures work well, but they don't provide a basis for determining the minimum order of a reduced order model for an "accurate" (user-defined) reproduction of the original system. The Moore algorithm does present such decision information. This will be discussed following the presentation of the four algorithm paths. Finally, the algorithm will be summarized and unique properties discussed.

Problem

For this study, only linear, time-invariant systems are considered, and in analysis of these systems, the following problem is common: given a system of high dimensionality described by n first order differential equations as:

$$\dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{u} \quad (1)$$

$$\underline{y} = \underline{C}\underline{x} + \underline{D}\underline{u} \quad (2)$$

where $\underline{x}$ is an n -vector ($n \times 1$)

$\underline{u}$ is a q -vector of inputs ($q \times 1$)

$\underline{y}$ is a m -vector of outputs ($m \times 1$)

(usually $n \gg m$ or q)

$\underline{A}$ is $n \times n$

$\underline{B}$ is $n \times q$

$\underline{C}$ is $m \times n$

$\underline{D}$ is $m \times q$

This form is known as the state variable representation of the system.

A system of the form:

$$\dot{\underline{x}}_r = A_r \underline{x}_r + B_r \underline{u} \quad (3)$$

$$\underline{y}_r = C_r \underline{x}_r + D \underline{u} \quad (4)$$

where $\underline{x}_r$ is a r -vector and $r \ll n$, is desired such that this low order model preserves the "important" input-output properties of the original system (1) - (2). These important properties include frequency response, transient and steady state response and that the resulting reduced order model remain at least as controllable and as observable as the original system.

Kalman Theory of Minimal Realization

Kalman uses Gilbert's canonical decomposition procedure to transform a state variable coordinate system into a coordinate system in which the system is clearly divided into completely controllable-completely observable (c.c. -c. o.), completely controllable-completely unobservable (c.c. -u. o.), completely uncontrollable-completely observable (u.c. -c. o.), and completely uncontrollable-completely unobservable (u.c. -u. o.) portions. Kalman states (Ref 19:169) that a system is irreducible if it consists only of the c.c. -c. o. portion of the original system. The input-output or transfer function properties of a system can only give information about the c.c. -c. o. representation

of a system. Another interesting result presented by Kalman is that a system is irreducible if the rank of the matrix of impulse response functions (the Hankel matrix) is of rank n . Unfortunately, however, the c.c. -c.o., c.c. -u.o., u.c. -c.o., and u.c. -u.o. subspaces cannot be determined accurately on a computer due to their numerical instability. Also the subspaces are highly state coordinate system dependent, meaning that for equivalent state space representations of a system, different subspaces exist. The Moore algorithm presents a way to yield a coordinate invariant, numerically stable method of obtaining these subspaces. In effect, the Moore algorithm provides "working subspaces" for the c.c. -c.o. subspace.

Presentation of the Moore Algorithm (Ref 24,25) (Appendix B)

The Moore algorithm finds a state space coordinate system which orders the states with respect to controllability and observability properties (i.e., most controllable-most observable states to the least controllable-least observable states). By stripping away the bottom states (i.e., the least controllable-least observable states) model order reduction is performed.

The controllability grammian (Equation 5) and the observability grammian (Equation 6) allow the c.c. -c.o. subspace to be numerically solved for

$$W_c^a(t) = e^{At} \left[\int_0^t e^{-A\tau} B B^T e^{-A^T \tau} d\tau \right] e^{A^T t} \quad (5)$$

$$W_O^a(t) = e^{A^T t} \left[\int_0^t e^{-A^T \tau} C^T C e^{-A \tau} d\tau \right] e^{A t} \quad (6)$$

The controllable subspace is

$$X_C = \text{Im} (W_C^a(t)) \quad (7)$$

The uncontrollable subspace is

$$X_{\bar{C}} = \text{Ker} (W_C^a(t))^T = \text{Ker} (W_C^a(t)) \quad (8)$$

The observable subspace is

$$X_O = \text{Im} (W_O^a(t))^T = \text{Im} (W_O^a(t)) \quad (9)$$

The unobservable subspace is

$$X_{\bar{O}} = \text{Ker} (W_O^a(t)) \quad (10)$$

Differentiating equations 5 and 6 respectively yields

$$\dot{W}_C^a(t) = A W_C^a(t) + W_C^a(t) A^T + [B B^T] \quad (11)$$

$$\dot{W}_O^a(t) = A^T W_O^a(t) + W_O^a(t) A + [C^T C] \quad (12)$$

Allowing $\dot{W}_C^a(t)$ and $\dot{W}_O^a(t)$ to be zero in the steady state produces the algebraic Lyapunov equations

$$0 = A W_C^a(t) + W_C^a(t) A^T + [B B^T] \quad (13)$$

$$0 = A^T W_O^a(t) + W_O^a(t) A + [C^T C] \quad (14)$$

By solving these equations for $W_C^a(t)$ and $W_O^a(t)$ yields the controllability and observability grammians respectively. By characterizing

the null space (kernel) and range space (image) of these grammians, the c.c. -c.o. subspace is identified.

However, as mentioned previously, these subspaces (grammians) are state coordinate system dependent. Therefore the Moore algorithm transforms the system into the internally balanced state coordinate system, $\underline{X} = T \underline{X}'$, using the linear transformation matrix

$$T = U_O \Sigma_O^{-1} U_H \Sigma_H^{\frac{1}{2}} \quad (\text{Ref 25, Appendix B}) \quad (15)$$

where U_O = left singular vectors of the observability grammian (for more information on singular values, vectors see Appendix A, Ref 36)

Σ_O = diagonal matrix containing singular values of the observability grammian

U_H = left singular vectors of the H_{INF} matrix

$$H_{INF} = \Sigma_O U_O^T U_C \Sigma_C$$

Σ_H = diagonal matrix containing singular values of the H_{INF} matrix

U_C = left singular vectors of the controllability grammian

Σ_C = diagonal matrix containing singular values of the controllability grammian

The Moore algorithm will now be presented. Each of the four main paths will be explained.

A flow chart illustrating the Moore algorithm is found in Figure 1. The reader is encouraged to refer to References 24, 25 and to Appendix B for the in-depth development.

Path 1 - Impulse Balancing. The algorithm flow-chart in Figure 1 is the "impulse balancing" algorithm. This algorithm yields its best results (i. e., lowest order models) for a system when subjected to an impulse input. This is not to say that reduced order models obtained via the impulse balancing method when responding to other classes of inputs (step, sine, etc.) will not respond well. The point is, the best replication of the original system for the lowest order model will occur if the model and the original system are responding to an impulse input.

Path 2 - Step Balancing. The algorithm is now modified to balance according to a step input. When this is accomplished, the reduced order models will be best for a step input.

The impulse response function (assuming no feedforward term) is

$$\underline{Y}(t) = C e^{At} B \quad (16)$$

The impulse is the time derivative of the unit step function (Ref 28:65).

Therefore, the step response is written

$$\underline{Y}(t) = C \left(\int_0^t e^{A\tau} d\tau \right) B \quad (17)$$

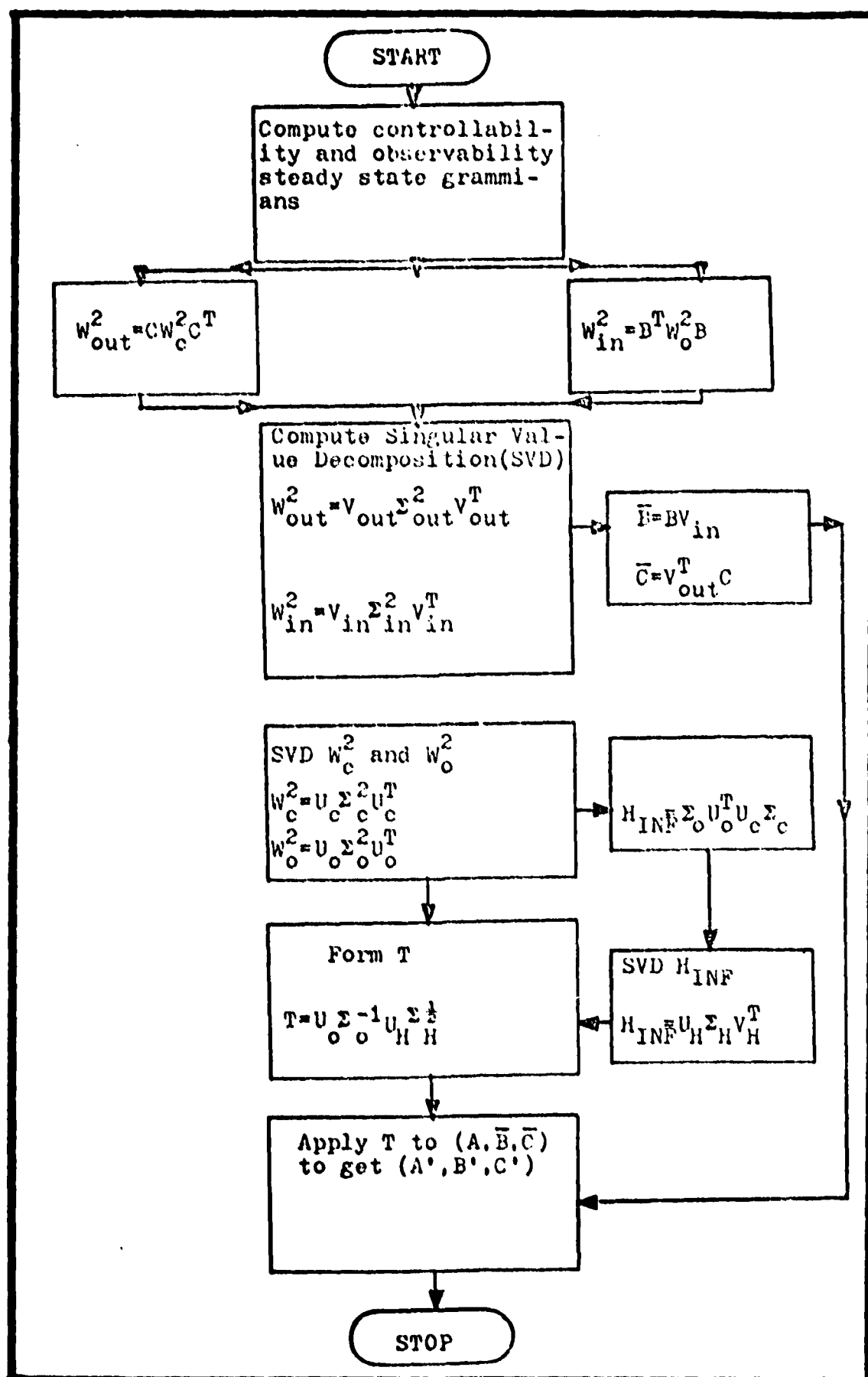


FIGURE 1. Flow Chart of Impulse Balancing Algorithm

Using the power series representation for e^{At} (Ref 29), equation (17) can be written

$$\underline{Y}(t) = C [e^{At} - I] A^{-1} B. \quad (18)$$

This is rearranged to produce

$$\underline{Y}(t) = C e^{At} (A^{-1} B) - C A^{-1} B. \quad (19)$$

Using the state space representation of the transfer function matrix (Ref 11:131),

$$G(s) = C [sI - A]^{-1} B, \quad (20)$$

and assuming no feedforward elements (no D matrix), and zero initial conditions, yields

$$\underline{Y}(s) = C [sI - A]^{-1} B U(s). \quad (21)$$

Applying the final theorem to equation (20) yields (Ref 22:103)

$$\underline{Y}(s) \Big|_{\substack{\text{steady} \\ \text{state}}} = -C A^{-1} B. \quad (22)$$

Since the second term in equation (19) is indeed the steady state part of the step response, then the first term is the transient portion.

Therefore,

$$\underline{Y}(t) \Big|_{\substack{\text{transient} \\ \text{(step input)}}} = C e^{At} (A^{-1} B). \quad (23)$$

The Moore algorithm gives results for the impulse response

$$\underline{Y}(t) = Ce^{At}B. \quad (24)$$

Therefore, if $(A, A^{-1}B, C)$ is input to the original balancing algorithm, it will yield the transient part of the step response rather than the impulse response. The steady state value differences between the reduced order model and the full order model are taken care of by adding a compensating D matrix to the reduced order system. This matrix is defined as

$$D_{COMP} = -[CA^{-1}B]_{FULL} - (-[CA^{-1}B]_{REDUCED}). \quad (25)$$

Using these two corrections, the algorithm now is a "step balancing" algorithm and utilizes the knowledge that the input is a step to provide lower order models for a step input than the impulse balancing algorithm can. Again, reduced order models providing "good" results can be obtained using both the impulse balancing and step balancing algorithms for other classes of inputs, but if the algorithm possesses knowledge of the type input, lower order models can usually be attained.

Path 3 - Infinite Interval Grammians.

Equations 13 and 14 presented the two matrix Lyapunov equations that yield the grammians for the interval between zero and

infinity. This tends to weight the steady state characteristics of the balanced design more heavily than the transient characteristics.

Path 4 - Finite Interval Grammians.

If the original integrals in equations 5 and 6 are numerically integrated from 0 to t , where t is some finite time, the steady state characteristics of the resulting model would not be as dominant as before. This path is not investigated in this thesis. However, through experimentation an "optimal" t could probably be found to "weight" the balanced model characteristics equally. This path is discussed further in Section VI in the Conclusions and Recommendations.

The H_{INF} Matrix

Most model order reduction techniques do not present a means for determining the lowest possible order that the model can be reduced to, thereby making the user perform many iterations to find this minimum order. The Moore algorithm does offer a guideline. The indicator stems from the H_{INF} matrix which is an integral part of the Moore balancing algorithm. (See equation (15) and the definition of terms following that equation.) The H_{INF} matrix is related to the discrete-time Hankel matrix (matrix of impulse responses) (Ref 16), where the Hankel matrix may be defined as the discrete time observability matrix multiplied by the discrete time controllability matrix.

$$H_{\text{ANKEL}} = \begin{bmatrix} C \\ CF \\ \vdots \\ CF^{n-1} \end{bmatrix} [B \quad FB \dots F^{n-1}B] = \begin{bmatrix} CB \quad CFB \dots CF^{n-1}B \\ CFB \\ \vdots \\ CF^{n-1}B \dots CF^{2(n-1)}B \end{bmatrix} \quad (26)$$

where $F = e^{AT}$ (T = sample time). The Hankel matrix is also identical to the matrix of impulse responses:

$$H_{\text{ANKEL}} = \begin{bmatrix} h(1) & \dots & h(N) \\ h(2) & & \vdots \\ \vdots & & \vdots \\ h(n) & \dots & h(2N-1) \end{bmatrix} \quad (27)$$

Then, taking the singular value factorization of the H_{INF} matrix and Hankel matrix, respectively, yields the following result:

$$\sigma_{H_{\text{INF}}}^i = \lim_{T \rightarrow 0} \sigma_{H_{\text{ANKEL}}}^i \quad (28)$$

where T is sampling time

$$\sigma = \text{singular value (eigenvalue of } H_{\text{INF}}^T H_{\text{INF}} \text{)}$$

Kalman in Reference 19 shows that the rank of the Hankel matrix (matrix of impulse response functions) gives information on whether a system is of minimum order. Moore uses the continuous-time, coordinate-invariant H_{INF} matrix to provide similar information.

This can best be understood by using the geometrical concept provided by the singular value factorization of the H_{INF} matrix (see Appendix A, B). The singular values of the H_{INF} matrix are the axis' lengths of the hyperellipsoid described by singular vectors of the H_{INF} matrix. The H_{INF} matrix (similar to the discrete-time Hankel matrix) has embedded in it controllability/observability properties. The "large" singular values are the axis' lengths of the major axes of the hyperellipsoid. These major axes span the controllable/observable subspace. Therefore, the "small" singular values correspond to the minor axes, and thus the uncontrollable/unobservable subspace. State redundancy is caused by uncontrollable/unobservable states (Ref 25). Therefore, the number of "large" singular values define the dimension of the controllable/observable (thus nonredundant) state space.

Example

An example is now presented which illustrates the state coordinate dependency of the controllability and observability grammians. The internally balanced representation will then be presented showing the coordinate invariant grammians (thus coordinate invariant subspaces).

The example has transfer function

$$G(s) = \frac{1}{(s+1+j2)(s+1-j2)(s+3)} \quad (29)$$

Three test cases were run. In each case the A matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -15 & -11 & -5 \end{bmatrix} \quad (30)$$

For case 1, the B and C matrices are

$$B_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C_1 = [1 \ 0 \ 0] \quad (31)$$

For case 2, the B and C matrices are

$$B_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \times 10^{-2} \end{bmatrix}, \quad C_2 = [1 \times 10^{+2} \ 0 \ 0] \quad (32)$$

For case 3, the B and C matrices are

$$B_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \times 10^{+2} \end{bmatrix}, \quad C_3 = [1 \times 10^{-2} \ 0 \ 0] \quad (33)$$

The impulse balanced system for impulse input for each case is

$$A' = \begin{bmatrix} .358 & .508 & 1.71 \\ .508 & -3.46 & 1.61 \\ 1.71 & -1.61 & -1.18 \end{bmatrix} \quad (34)$$

$$B' = \begin{bmatrix} .202 \\ -.152 \\ .253 \end{bmatrix}, \quad C' = [.202 \ -.152 \ -.253]$$

The step balanced system (Moore balanced system for step input) for each case is

$$A' = \begin{bmatrix} 1.19 & .596 & 1.73 \\ .596 & -2.87 & 1.06 \\ -1.73 & -1.06 & -.941 \end{bmatrix} \quad (35)$$

$$B' = \begin{bmatrix} .137 \\ -.532 \\ .550 \end{bmatrix}, \quad C' = [.286 \quad -.074 \quad -.142]$$

The respective controllability and observability grammians are shown in Table I. This table shows the considerable variance in the magnitude of the elements in the controllability and observability grammians for Cases 1, 2, 3. The grammians shown for the impulse balancing and the step balancing algorithms are unique for the given system.

TABLE I. Test Case Controllability and Observability Grammians

CASE	CONTROLLABILITY GRAMMIAN	OBSERVABILITY GRAMMIAN
1	$\begin{bmatrix} 4.2E-3 & 1.8E-12 & -1.3E-2 \\ 1.8E-12 & 1.3E-2 & 1.6E-11 \\ -1.3E-2 & 1.6E-11 & 1.4E-1 \end{bmatrix}$	$\begin{bmatrix} 6.8E-1 & 2.3E-1 & 3.3E-3 \\ 2.3E-1 & 1.2E-1 & 2.1E-2 \\ 3.3E-2 & 2.1E-2 & 4.2E-3 \end{bmatrix}$
2	$\begin{bmatrix} 4.2E-7 & 1.8E-16 & -1.3E-6 \\ 1.8E-16 & 1.3E-6 & 1.6E-15 \\ -1.3E-6 & 1.6E-15 & 1.4E-5 \end{bmatrix}$	$\begin{bmatrix} 6.8E+3 & 2.3E+3 & 3.3E+2 \\ 2.3E+3 & 1.2E+3 & 2.1E+2 \\ 3.3E+2 & 2.1E+2 & 4.2E+1 \end{bmatrix}$
3	$\begin{bmatrix} 4.2E+1 & 1.8E-8 & -1.3E+2 \\ 1.8E-8 & 1.3E+2 & 1.6E-7 \\ -1.3E+2 & 1.6E-7 & 1.4E+3 \end{bmatrix}$	$\begin{bmatrix} 6.8E-5 & 2.3E-5 & 3.3E-6 \\ 2.3E-5 & 1.2E-5 & 2.1E-6 \\ 3.3E-6 & 2.1E-6 & 4.2E-7 \end{bmatrix}$
Impulse Balancing	$\begin{bmatrix} 5.7E-2 & 1.3E-7 & -3.7E-7 \\ 1.3E-7 & 3.3E-3 & 3.0E-7 \\ -3.7E-7 & 3.0E-7 & 2.7E-2 \end{bmatrix}$	$\begin{bmatrix} 5.7E-2 & 8.2E-7 & 2.0E-7 \\ 8.2E-7 & 3.3E-3 & -7.4E-7 \\ 2.0E-7 & -7.4E-7 & 2.7E-2 \end{bmatrix}$
Step Balancing	$\begin{bmatrix} 8.1E-2 & -6.1E-3 & 5.2E-2 \\ -6.1E-3 & 3.2E-2 & -4.3E-2 \\ 5.2E-2 & -4.3E-2 & 1.1E-1 \end{bmatrix}$	$\begin{bmatrix} 3.4E-2 & 6.2E-8 & 3.8E-8 \\ 6.2E-8 & 9.4E-4 & -1.4E-7 \\ 3.8E-8 & -1.4E-7 & 1.1E-2 \end{bmatrix}$

The singular values of the H_{INF} matrix for impulse balancing are

$$\begin{aligned}\sigma_1 &= 5.7226 \times 10^{-2} \\ \sigma_2 &= 2.7235 \times 10^{-2} \\ \sigma_3 &= 3.3416 \times 10^{-3}\end{aligned}\tag{36}$$

The singular values of the H_{INF} matrix for step balancing are

$$\begin{aligned}\sigma_1 &= 3.4278 \times 10^{-2} \\ \sigma_2 &= 1.0774 \times 10^{-2} \\ \sigma_3 &= 9.4013 \times 10^{-4}\end{aligned}\tag{37}$$

As mentioned, these singular values provide a guideline for how low the system order can be reduced. For both the impulse balancing and step balancing paths, the singular values of the H_{INF} matrix seem to cluster. There seems to be two "large" singular values in each case. This indicates that this system probably should not be reduced below second order for an "accurate" representation of the original system.

The H_{INF} matrix coupled with the four paths in the Moore algorithm provide options that other model order reduction techniques do not possess.

Summary of the Moore Algorithm and Internally Balanced Coordinate System

The Moore algorithm is implemented with numerically stable code. The code uses readily available subroutine packages such as IMSL, EISPACK, and LINPACK (Ref 13).

Because of this numerically stable code, and because of the linear transformation (Equation 15) to a coordinate invariant state representation, the Moore algorithm provides "working subspaces" for the Kalman decomposition.

The resulting internally balanced state coordinate system orders the states with respect to controllability and observability properties. Therefore, deletion of the bottom states of the internally balanced representation is in effect stripping away the uncontrollable, unobservable (thus redundant) subspace.

The H_{INF} matrix provides a guideline for the selection of the minimum order for the resulting reduced order model. This feature does not exist with most other model order reduction techniques.

The algorithm also minimizes the maximum of the condition number of the controllability and observability grammians. This property indicates that the balanced system is more robust to perturbations in system parameters.

The algorithm has now been presented. The next step is to evaluate the algorithm by applying it to real world systems. But before this can be accomplished, a tool must be developed which possesses the capabilities to investigate the reduced order model in both the time domain and frequency domain. In addition, it will be desirable to investigate the singular values of the H_{INF} matrix and

the controllability and observability grammians. The next section presents an interactive software package which accomplishes the above analysis objectives.

III. DEVELOPMENT OF INTERACTIVE SOFTWARE INTRODUCTION

INTRODUCTION

In this section, an interactive computer program is developed that provides options useful in the study of multi-input-multi-output systems. The program is appropriately called MIMO. MIMO was developed in addition and in parallel to the application of the Moore algorithm presented in the previous sections.

Several options incorporated in MIMO are direct results from the study performed on the Moore algorithm. The remaining options yield various "special" state coordinate systems, each of which possesses special properties. A user's manual for MIMO is contained in Appendix C. Appendix D contains the Fortran listing of MIMO.

The structure of the program MIMO will be presented first. A description of the functions performed by each option will follow. The reader is encouraged to refer to the user's manual for an explanation on how to access and use MIMO, for that information will not be discussed here.

PROGRAM STRUCTURE

Several objectives influenced the design of MIMO. MIMO is an interactive program and is restricted to 60K (octal) words of memory. The program is designed to be readily modified. Individual options may be easily added, deleted or changed. MIMO

interfaces to TOTAL (Ref 21) for time domain and frequency domain responses to be obtained. In the user's manual, efforts are made to explain how to prevent premature termination of MIMO due to an input error. However, time precludes the incorporation of an extensive error checking capability into the program itself. The main emphasis is instead to provide the user with accurate, efficient options and not at this time with an "optimized" user interface.

MIMO utilizes overlays (Ref 8, 10) to meet the memory restrictions and to provide a degree of modularity to its design. Overlays divide a large program into a set of smaller programs, each of which is loaded into memory if and when it is required. The loading of these smaller programs is controlled by an executive overlay which remains in memory at all times. The smaller programs which may be referred to as primary overlays, may be subdivided into secondary overlays if necessary. Thus, the largest amount of memory ever required is the sum of the executive overlay's memory requirement and the sum of the largest primary and secondary overlays' memory requirements. Since only one primary overlay can reside in memory with the executive overlay at any given time, the modularity of the program is achieved. Additional options become additional primary overlays. Any sub-functions peculiar to any one option may be made secondary overlays. A more extensive discussion of overlays is found in Reference 8.

The additional objective, that MIMO interface to TOTAL (Ref 21), is met by storing all pertinent information in mass storage on a file named "MEMAUX". TOTAL also uses a file named MEMAUX. By writing the appropriate information with the appropriate index, the interface is achieved. The exact procedure for accomplishing this objective is found under Option 12 in Appendix C.

MIMO allows the user to store all current, pertinent information in the local file "MEMORE". This allows the user to terminate MIMO, execute other interactive computer functions, return to MIMO, and recover to the point of departure from the program.

MIMO contains two options which produce plots. A high degree of flexibility with respect to plots is achieved by MIMO. The user may dispose his accumulated plots to the AFIT terminal automatically with Option 9. He may also perform the standard routes, save the plot file, or use TEK PLOT or CCPREV (see AFIT OCR). Extensive descriptions of these procedures may be found in the user's manual in Appendix C.

With these objectives as well as restrictions incorporated, MIMO's options were developed. The description of each option to the extent that it does not duplicate the information contained in the user's manual is now presented.

PRESENTATION OF OPTIONS

MIMO's options work exclusively with matrices. Correspondingly, only the state variable description of a dynamic system is utilized. To meet computer memory restrictions, matrices are limited to 10 x 10's. The controllability matrix is limited to a 10 x 30, the observability matrix to a 30 x 10, and the Hankel matrix to a 30 x 30. At this time, sixteen options exist in MIMO. Their descriptions follow.

Option 0 lists the available options. The list will be amended as additional options are added.

Option 1 causes MIMO to terminate. All pertinent information is automatically saved in the local file MEMORE.

Input of the state space matrices (A,B,C) is accomplished in the first part of Option 2. After input is accomplished, the user may cease execution of Option 2, or obtain the "impulse balanced", or the "step balanced" systems obtained via the Moore algorithm. Using these balanced systems, reduced order models are obtained by deleting the bottom states (see Appendix C, Option 2) (Ref 24, 25).

Option 3 uses a truncated power series approximation to obtain the (F,G,C) discrete-time system for a given value of sample time, from the (A,B,C) system input in Option 2 (valid for a linear, time invariant system only). The discrete-time controllability, observability, and Hankel matrices (Ref 28) are also obtained. The user may

optionally suppress the listing. This is useful, i.e., if the user wishes to obtain the Hankel matrix and proceed to Option 7 to obtain its singular values/singular vectors.

The power series used in Option 3 is now presented. For a linear time invariant system,

$$F = e^{AT} \text{ (where } T = \text{ sampling time).} \quad (38)$$

The G matrix is defined by

$$G = \int_0^T e^{A\tau} d\tau B \quad (39)$$

The power series representation for e^{AT} is (Ref 28:37),

$$e^{AT} = I + AT + \frac{A^2 T^2}{2!} + \dots \quad (40)$$

The power series representation for $\int_0^T e^{A\tau} d\tau$ is

$$\int_0^T e^{A\tau} d\tau = I T + \frac{AT^2}{2!} + \dots \quad (41)$$

Therefore, avoiding having to take A^{-1} , e^{AT} may be represented as

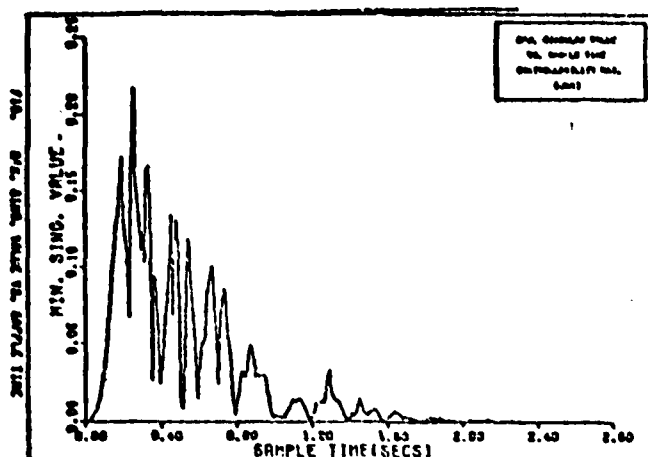
$$e^{AT} = A * \left[\int_0^T e^{A\tau} d\tau \right] + I \quad (42)$$

MIMO truncates this power series after 50 terms. The approximation is accurate if the magnitude of the largest eigenvalue of A multiplied by sample time is less than one (Ref 28:35). If this condition is not satisfied, convergence of the power series may not occur.

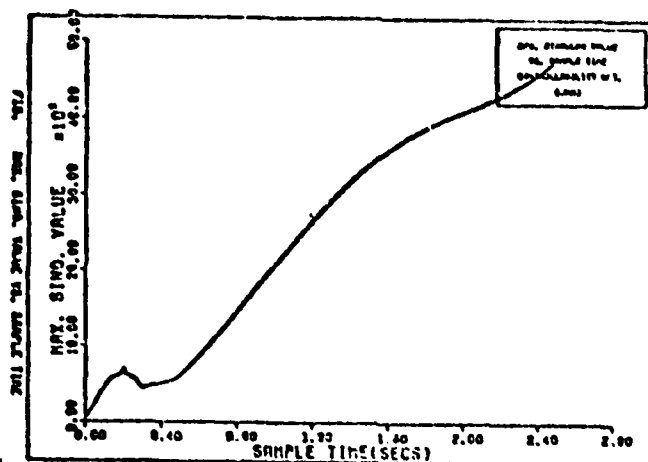
Option 4 plots and optionally lists singular values of the discrete time controllability, observability and/or Hankel matrices versus sample time. The user may list all singular values for each matrix for each sample time in a user-specified range, or he may just plot the singular values. The reader should refer to the user's manual for information on how to accomplish this option.

This option has several applications. The plot of the condition number ($\sigma_{\max}/\sigma_{\min}$) versus sample time, of the Hankel matrix provides an excellent criterion for choice of sample time for a discrete-time system (Ref 30). Plots of the minimum singular values of the controllability and observability matrices provide a measure of the degree of control, or degree to which a system may be observed respectively.

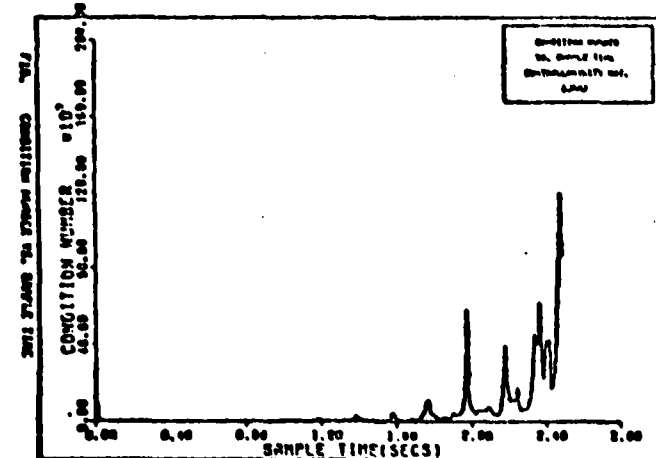
An example of this option as applied to the B-52E flutter control problem's twenty-fourth order system is now presented. All of the nine possible plots were generated. The B-52E model reveals that for the dominant eigenvalues, the period is .4051 seconds. Reid (Ref 30) shows that any integer multiple of this period would be a poor choice of sample time for a discrete time representation of the continuous-time system. The plots verify this fact. Figure 2 shows that the minimum singular value tends toward zero at integer multiples of .4051. As this singular value approaches zero, the



a. Minimum Singular Value



b. Maximum Singular Value



c. Condition Number ($\text{sig}(\text{max})/\text{sig}(\text{min})$)

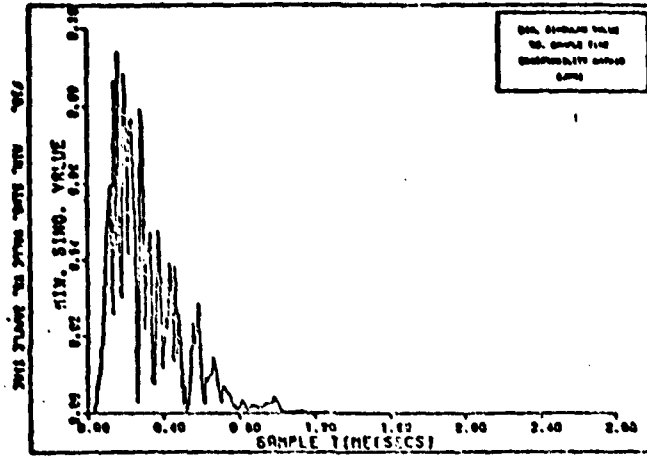
FIGURE 2. Controllability Matrix Max. and Min. Singular Values and Condition Number vs. Sample Time

controllability matrix is "closer" to singularity. This indicates that for an integer multiple of .4051, the degree of control on the system is lessened. Figure 3a shows that the observability matrix minimum singular value also tends toward zero as the sample time approaches an integer multiple of .4051. This indicates that the system is less observable at these sample times.

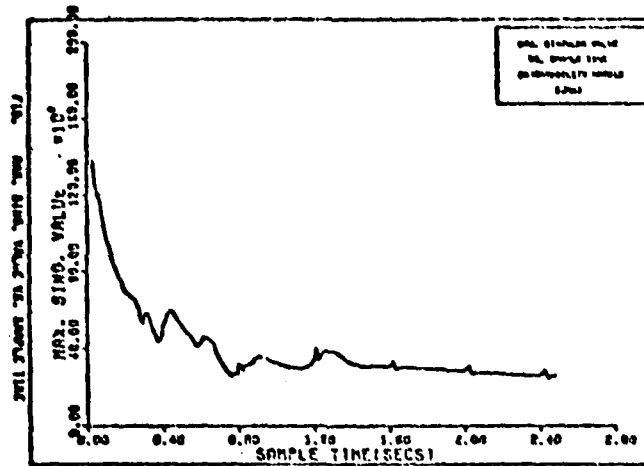
Perhaps though, Figure 4c is the most important. Clearly, the condition number of the Hankel matrix is maximized at integer multiples of .4051. This indicates that the system is most susceptible to perturbations (model mismatch/truncation, wordlength restrictions, etc.) in system elements. A "good" choice of sample time for a system then would be one that minimizes the condition number (K) of the Hankel matrix (maximized the reciprocal condition number in Reid's papers).

This option provides very revealing and helpful information in the choice of sample time for a system. For more information, see Reference 30.

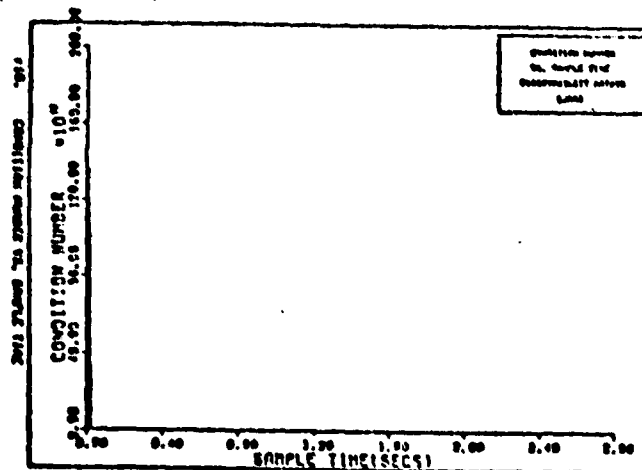
Option 5 obtains the continuous time controllability and observability matrices for the (A,B,C) system input in Option 2. This option is useful if one desires to check the controllability and/or observability of his particular representation of a system. He may first input the system in Option 2, proceed to Option 5 to obtain these



a. Minimum Singular Value

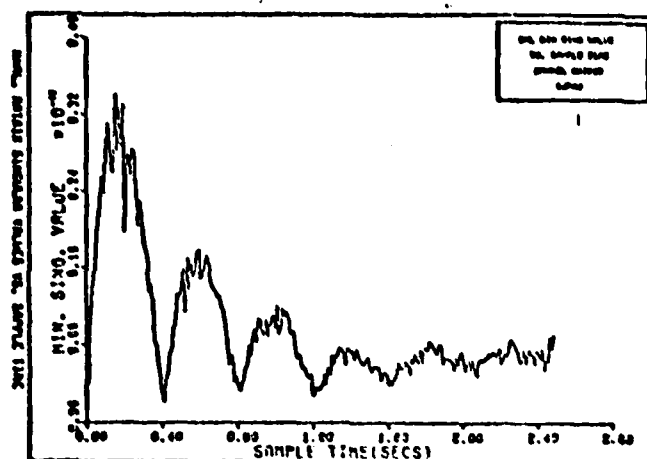


b. Maximum Singular Value

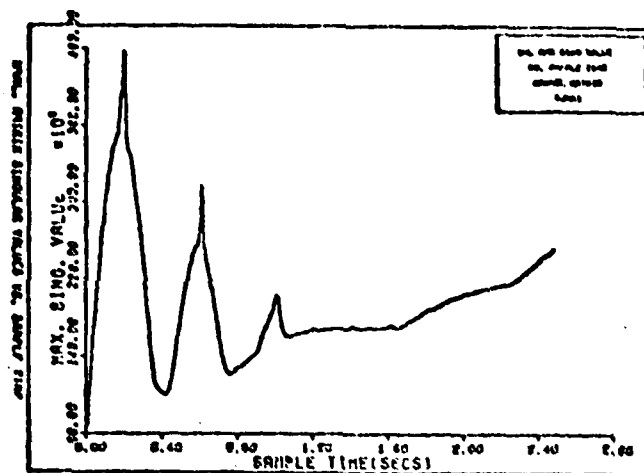


c. Condition Number ($\text{sig}(\text{max})/\text{sig}(\text{min})$)

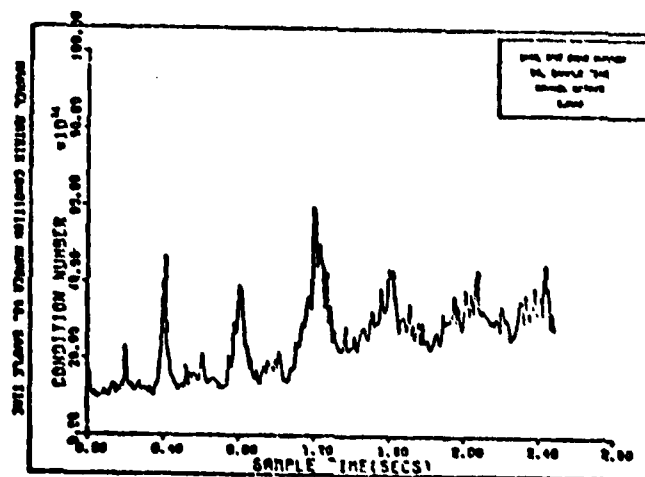
FIGURE 3. Observability Matrix Max. and Min. Singular Values and Condition Number vs. Sample Time



a. Minimum Singular Value



b. Maximum Singular Value



c. Condition Number ($\text{sig}(\text{max})/\text{sig}(\text{min})$)

FIGURE 4. Hankel Matrix Max. and Min. Singular Values and Condition Number vs. Sample Time

matrices, and then go to Option 7 to obtain the singular values of the controllability and/or observability matrices. Generally, any singular value of magnitude less than 1.0×10^{-6} can be interpreted as zero. Therefore, the rank of the controllability and/or observability matrices is determined by the number of non-zero singular values.

Option 6 implements a portion of LINPACK (Ref 13) to estimate the condition number of a square, real matrix. This method as developed in Reference 7 is a highly efficient method for obtaining the condition number with respect to inversion. This method does not require the relatively expensive singular value decomposition. Instead, the condition number can also be defined as

$$K = \|A^{-1}\| \|A\| \quad (43)$$

where A is a real matrix

$\|\cdot\|$ denotes the subordinate spectral norm for matrix, i.e.,

$\|\cdot\|_2$ denotes the standard Euclidean (l_2) norm for a vector.

Using this definition, the algorithm yields a highly accurate estimate of the condition number with respect to inversion.

An example is now presented. One of the most ill-conditioned matrices with respect to inversion is a Hilbert matrix. A Hilbert matrix is of the form.

$$H_{IL} = \begin{bmatrix} 1/2 & 1/3 & \dots & 1/N+1 \\ 1/3 & 1/4 & \dots & 1/N+2 \\ \vdots & & & \vdots \\ 1/N+1 & \dots & & 1/2N \end{bmatrix}_{N \times N} \quad (55)$$

As the order of the system increases, the estimate is more ill conditioned. In Table II the estimate of the condition number with respect to inversion is compared with $\sigma_{\max}/\sigma_{\min}$, obtained via the matrix's singular value decomposition. The results are quite good. Therefore, when the condition number of a matrix is desired, Option 6 is an efficient way of producing that result via the LINPACK algorithm (Ref 7, 13) for most applications.

TABLE II. Comparison of LINPACK
Estimate of Condition Number Vs. $\sigma_{\max}/\sigma_{\min}$
for a Hilbert Matrix

<u>Matrix Order</u>	<u>Option 6 Estimate</u>	<u>$\sigma_{\max}/\sigma_{\min}$</u>	<u>DIF</u>
2	43.097	37.836	5.261
3	545.691	441.952	103.738
5	329.236	238.498	90.737

Option 7 obtains the singular values and optionally the left and right singular vectors for a real, rectangular matrix. The particular matrix can be: A matrix input at this time, the original A matrix, the internally balanced A matrix, the discrete time F matrix, or the discrete time Hankel matrix.

The singular value factorization of a real, rectangular matrix yields (see Appendix A, Ref 36):

$$A_{MXN} = U_{MXM} \Sigma_{MXP} V_{NXN}^T \quad (45)$$

where U is a matrix whose columns are called the left singular vectors.

Σ is a diagonal matrix whose diagonal elements are called the singular values and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq \sigma_{r+1} = 0$.

$$r = \min (M, N)$$

V is a matrix whose columns are called the right singular vectors.

To present the multiple uses of this factorization is beyond the scope of this thesis. The reader is referred to References 13, 24, 25, 28, 36, or 30 for excellent presentations on the subject.

Option 8 plots and optionally lists the magnitude and phase responses of a user specified reduced order system. The method used reduces the amount of computations as required by conventional algorithms and utilizes highly efficient eigenvalue - eigenvector routines to obtain its results.

The derivation follows (Ref 28):

$$G(jw) = F \{ C e^{At} B \} \quad (46)$$

$$= F \left\{ \sum_{i=1}^N C(\underline{V}_i) (\underline{W}_i^T B) e^{\lambda_i t} \right\} \quad (47)$$

where $F \{ \cdot \}$ denotes the Fourier transform

$$[\underline{V}_1 \ \underline{V}_2 \ \dots \ \underline{V}_n] = \text{matrix of eigenvectors}$$

$$[\underline{W}_1 \ \underline{W}_2 \ \dots \ \underline{W}_n]^T = [\underline{V}_1 \ \underline{V}_2 \ \dots \ \underline{V}_n]^{-1}$$

λ_j denotes the j th eigenvalue

Now, rewriting Equation 47 yields:

$$G(jw) = \sum_{i=1}^N (C \underline{V}_i) (\underline{W}_i^T B) \frac{1}{jw + \lambda_i} \quad (48)$$

Using Equation 48 above, the magnitude and phase of the system for a given value of w can easily be obtained.

Option 9 disposes accumulated plots created by Option 4 and/or Option 8 to the AFIT terminal. This option uses an AFIT subroutine written by Captain Jerry Stinson (AFIT OCR). It also uses two subroutines contained in the BATELLE package (see AFIT OCR).

The program can easily be modified to send the plots to any remote terminal with an on-line Calcomp plotter at Wright-Patterson Air Force Base, Ohio.

Option 10 writes all pertinent information to local file MEMORE. Option 10 is automatically called when Option 1 is executed.

Option 11 recovers all pertinent information from the local file MEMORE. This enables the user to recover to the point in MIMO where he used Option 1 or Option 10.

Option 12 interfaces to TOTAL (Ref 26). The original (A,B,C) system, and the internally balanced (A',B',C') system are written to the mass storage file MEMAUX. Appendix C, Option 12 explains how to access TOTAL and use the information transferred to TOTAL from MIMO. TOTAL (Ref 21) is an interactive program which produces time domain and frequency domain responses for a transfer function with a defined input. Numerous options useful to the control system designer are available. Therefore, the interface to TOTAL from MIMO saves the user from having to retype large matrices.

Option 13 generates the output normal state coordinate system (Ref 24, 25) from the internally balanced system obtained for the original input system. This system is a byproduct of the Moore algorithm. The system is now balanced with respect to observability properties only. The states are ordered from the most observable to the least observable. This option is limited to SISO systems only.

Option 14 produces the output predictive state coordinate system (Ref 30, 29). The transformation:

$$\underline{X} = M_{OBS}^{-1} \bar{\underline{X}} \quad (49)$$

where M_{OBS} is the observability matrix yields:

$$\bar{M}_{OBS} = I \quad (50)$$

Reid shows in Reference 30 that:

$$\underline{Y} (k+n/k-1) = \begin{bmatrix} C \\ CF \\ \vdots \\ CF^{n-1} \end{bmatrix} \quad (51)$$

$$\text{where } \underline{Y} (k+n/k-1) = \begin{bmatrix} y (k+n/k-1) \\ y (k+n+1/k-1) \\ \vdots \\ y (k+2n-1/k-1) \end{bmatrix}$$

where $\underline{Y} (k+n/k-1)$ denotes the vector of predicted outputs, to n samples in the future based upon inputs up to the current time k . By transforming to the output predictive state coordinate system, the following equation results

$$\underline{Y} (k+n/k-1) = \bar{F}^n \bar{\underline{X}} (k) \quad (52)$$

where $\bar{F}^n = M_o^{-1} F M_o$

M_o = observability matrix

$F = e^{AT}$

This implies that the outputs at discrete times in the future based on inputs up to time K may be determined by multiplying the current state vector $\underline{X}(K)$ by $\bar{F}^n$ (where $\bar{F}$ is now in the "output predictive" state coordinate system). See Reference 30 for uses of this coordinate system.

Finally, Option 15 obtains the steady state controllability and observability grammians for the (A, B, C) system input in Option 2.

The controllability grammian is defined as:

$$W_c^a(O, t_1) = e^{At} \int_0^{t_1} e^{At} B B^T e^{A^T t} dt e^{A^T t} \quad (53)$$

The observability grammian is defined as:

$$W_o^a(O, t_1) = e^{A^T t} \int_0^{t_1} e^{A^T t} C^T C e^{At} dt e^{At} \quad (54)$$

Differentiating the two grammians respectively yields the two Riccati equations

$$\dot{W}_c^a(t) = A W_c^a(t) + W_c^a(t) A^T + B B^T \quad (55)$$

$$\dot{W}_o^a(t) = A^T W_o^a(t) + W_o^a(t) A + C^T C \quad (56)$$

In the steady state, $\dot{W}_c^a$ and $\dot{W}_o^a$ equal zero. Therefore equations 55 and 56 become respectively:

$$A W_C^a(t) + W_C^a(t) A^T + BB^T = 0 \quad (57)$$

$$A^T W_O^a(t) + W_O^a(t) A + C^T C = 0 \quad (58)$$

Note, that the underlying (A, B, C) system must be stable because the modified Bartels-Stewart algorithm that solves the required matrix Ricatti equations must have negative eigenvalues to work with. This author believes newer algorithms exist that surmount this problem, but has not been able to locate such. The interested reader should refer to Reference 20 as the possible source of new algorithms that solve matrix Lyapunov equations.

This concludes the presentation of the interactive program MIMO. All goals strived for were met. The program is interactive (60K(octal) words of memory are required), uses overlays which provide a modularity and ease of modification to the program, interfaces to TOTAL (Ref 21) as desired, and although as yet doesn't possess an extensive error checking capability, does utilize the most efficient algorithms currently available to provide the user with a unique and hopefully beneficial analysis tool.

IV. APPLICATION OF MOORE ALGORITHM TO SISO EXAMPLE

Introduction

In Section II the Moore algorithm was presented and its potential for application to real world systems discussed. However, it was pointed out that an analysis tool would be required to investigate the actual performance of the Moore algorithm on real-world problems. Section III presented such a tool. MIMO coupled with TOTAL and some batch software now provide the means by which the Moore algorithm will be investigated. However, before the algorithm is applied to a large dimension, real world system, a third order SISO example will be presented which illustrates the application of the Moore algorithm, utilizes MIMO and TOTAL for analysis purposes, and compares the Moore algorithm with three other model reduction algorithms.

This section will be organized in the following manner. First, a brief discussion of the other three algorithms will be presented to familiarize the reader with the model order reduction approaches compared with Moore's algorithm for this example. Then, the third order example will be presented. A summary of the section will follow.

Description of Other Algorithms

Though the Moore algorithm works with the state space representation of a system, the three algorithms to be presented here use

the system transfer function for their calculations. The transfer function matrix is related to state space (assuming no feedforward) via:

$$G(s) = C [sI - A]^{-1} B \quad (59)$$

T. Shamash (Ref 33) utilizes the Routh stability criterion and the Padé approximation to yield reduced order models. His basic procedure follows. Given:

$$G(s) = \frac{d_1 s^{n-1} + d_2 s^{n-2} + \dots + d_n}{e_0 s^n + e_1 s^{n-2} + \dots + e_n} \quad (60)$$

compute:

$$\hat{G}(s): \hat{G}(s) = \frac{1}{s} G\left[\frac{1}{s}\right] = \frac{d_n s^{n-1} + \dots + d_1}{e_n s^n + e_{n-1} s^{n-2} + \dots + e_0} \quad (61)$$

Once this has been accomplished, $\hat{G}(S)$ is expanded into a continued fraction as:

$$\begin{aligned} \hat{G}(s) = & \frac{1}{1 + \alpha_1 s} + \frac{1}{\alpha_2 s} + \dots + \frac{1}{\alpha_n s} \quad [B_1] \\ & + \frac{1}{\alpha_2 s} + \frac{1}{\alpha_3 s} + \dots + \frac{1}{\alpha_n s} \quad [B_2] \\ & + \dots \\ & \dots \frac{1}{\alpha_n s} \quad [B_n] \dots \end{aligned} \quad (62)$$

where

$$\frac{1}{1 + \alpha_1 S} + \frac{1}{\alpha_2 S} + \dots + \frac{1}{\alpha_n S} = \frac{1}{1 + \alpha_1 S + \frac{1}{\frac{\alpha_2 S + 1}{\alpha_3 S + 1} \dots \frac{1}{\alpha_n S}}} \quad (63)$$

This procedure allows the algorithm to obtain the Routh coefficients from the α_i 's and B_i 's in Equations 62 and 63. Then, for the K th order transfer function (where $K \ll N$), the first K terms are summed to yield $R(S)$. The reciprocal relationship as used in Equation (61) is utilized to produce $R(S)$, where $R(S)$ is of lower order than the original $G(S)$. This is a simplified presentation of the Shamash algorithm. In fact, there are many more computations used to efficiently obtain the coefficients of $R(S)$.

The Shamash algorithm has many desirable properties. It preserves original system stability and can be applied to multivariable and discrete-time systems. However, it uses the transfer function f or its computations of the reduced order transfer function. For multivariable systems, especially, the storage requirements get to be huge. Thus the computational burden is increased by the use of transfer functions versus state space for a fairly large order, multivariable system.

El Attar and Vidyasagar (Ref 14) have developed an algorithm which treats the impulse response (or transfer function) as an input-output map. The procedure consists of minimizing one of two possible functionals. The choice of functional is dependent on the type of input used. In effect, the algorithm gives a best possible uniform approximation for the given input. This algorithm is useful when the poles of a system are not widely separated (i. e. , a nondominant set of poles may not exist). In addition, original system stability is preserved and the algorithm is applicable to discrete-time systems, MIMO systems, and unstable systems. The algorithm, however, is by necessity input dependent.

Reddy (Ref 27) minimizes the integral of the square of the error between the corresponding real and imaginary parts of the original and assumed transfer functions' numerator and denominator to thereby evaluate the proper coefficients for the simplified transfer function. This algorithm is only applicable to SISO systems and does not ensure that the reduced order transfer function is stable, for a stable, original transfer function.

SISO Example

The following third order SISO example is presented as a comparison between the Moore algorithm and three other algorithms discussed in the previous section. The results of the model reduction for these other three algorithms are taken from Ref. 14. The results

for the Moore algorithm are obtained via use of MIMO. The unit impulse responses and frequency responses are obtained for Moore's second order model, as well as for El-Attar's (Ref 14), Shamash's (Ref 33), and Reddy's (Ref 27) second order models.

The original third order transfer function is:

$$GH(s) = \frac{1}{(s + 0.99)(s + 1.0)(s + 1.1)} \quad (64)$$

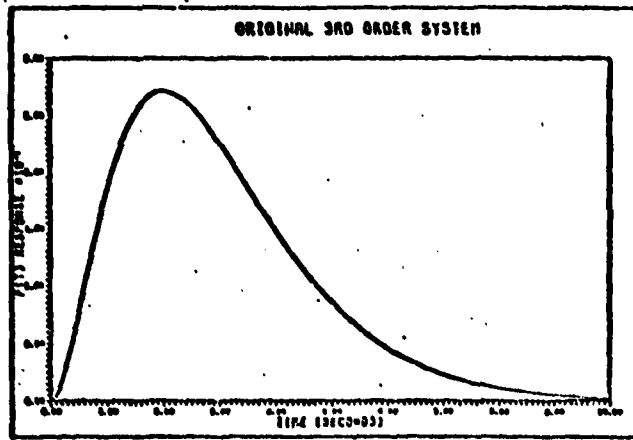
El Attar minimizes functions of induced norms. Shamash uses Pade' approximants and the Routh Stability criterion. Reddy minimizes an error criterion which is a function of the difference between the full order transfer function and the reduced order transfer function. Reddy's procedure results in an unstable second order model. The resulting reduced order transfer functions are shown in Table III.

The impulse response to a unit impulse input of the original system is shown in Figures 5a, 6a. Moore's second order system in response to the unit impulse is shown in Figure 5b. El-Attar's second order system in response to a unit impulse is in Figure 5c. El-Attar's and Moore's systems seem to be very close. This is true because El-Attar is using induced norm characteristics, while Moore's algorithm, based primarily upon the singular value factorization (where $\sigma_{\max} = \|A\|$, A is a general matrix) (see Appendix A), is directly related to induced norms. Shamash's and Reddy's second order system's unit impulse response responses are found in Figures 6b and 6c, respectively.

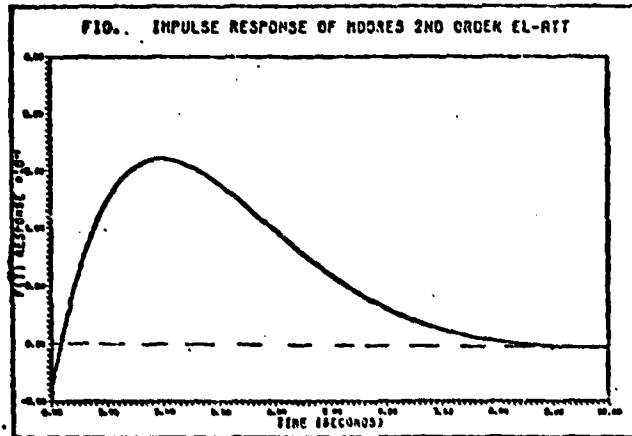
TABLE III

Reduced Order Model Transfer Functions for SISO Example

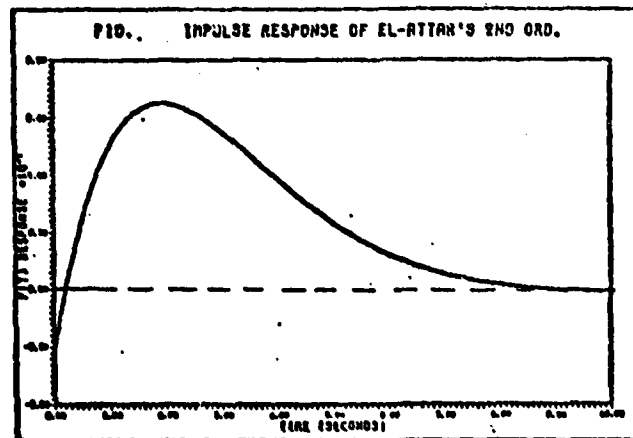
ALGORITHM	TRANSFER FUNCTION
1. Moore	$\frac{-.4462s+.5870}{s^2+.09971s+.8827}$ $= \frac{-.4462(s-1.316)}{(s+.04985-j.9382)}$
2. El-Attar Vidyasagar	$\frac{-.0911s+.3806}{s^2+1.0054s+.3869}$ $= \frac{-.0911(s-4.1778)}{(s+.5027-j.3663)}$
3. Shamash	$\frac{.00022s+.3746}{s^2+1.12s+.3708}$ $= \frac{.00022(s+1702.73)}{(s+.56-j.2391)}$
4. Reddy	$\frac{1}{2.5s^2-1.34845s+1}$ $= \frac{.4}{(s-.620)(s+.080)}$



a. Original 3rd Order System

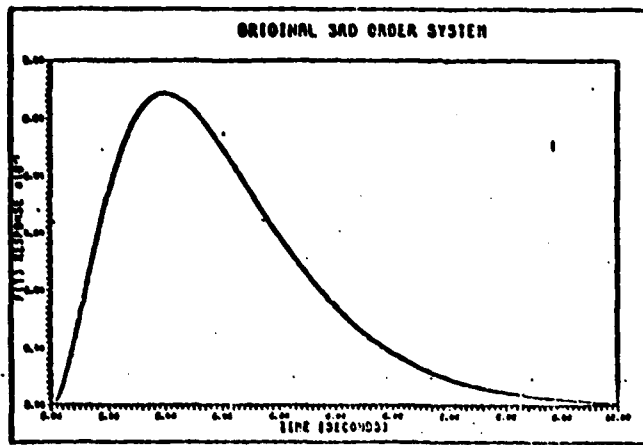


b. Moore's 2nd Order System

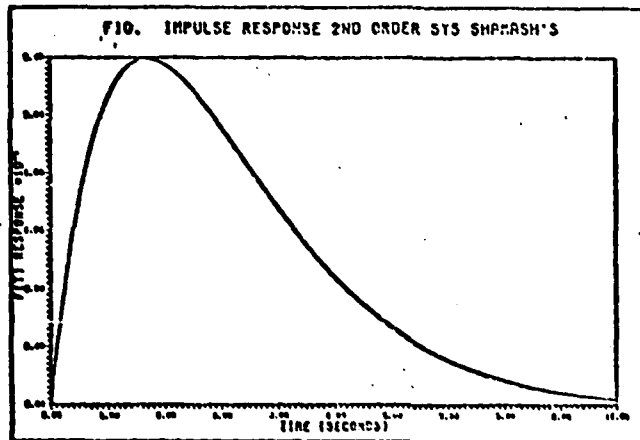


c. El-Attar's 2nd Order System

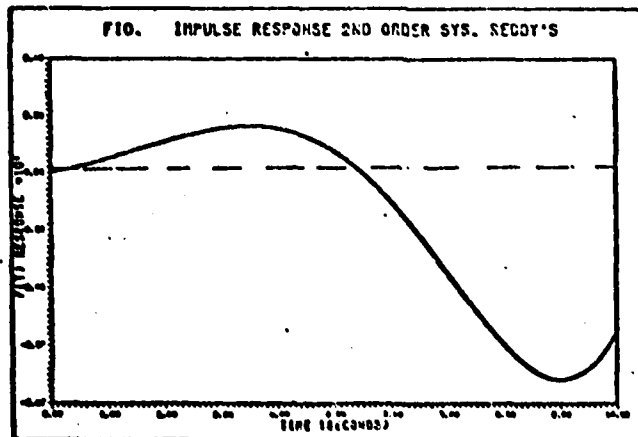
FIGURE 5. Unit Impulse Responses for SISO Example



a. Original 3rd Order System

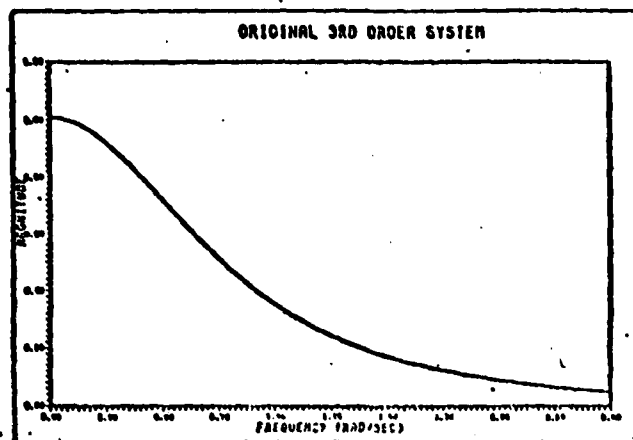


b. Shamash's 2nd Order System

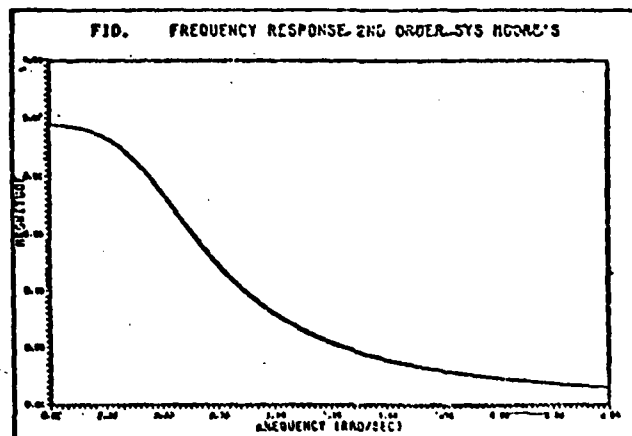


c. Reddy's 2nd Order System

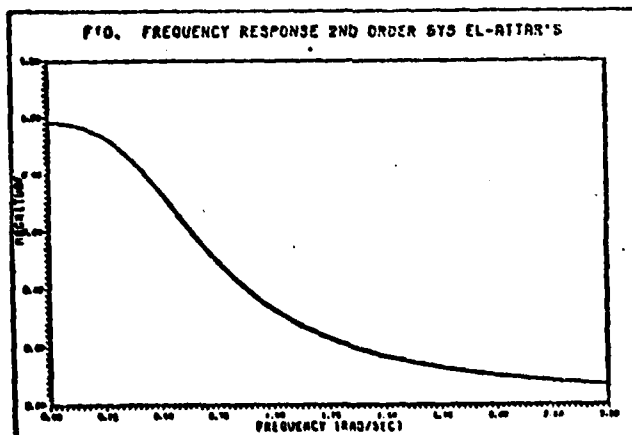
FIGURE 6. Impulse Responses for SISO Example



a. Original 3rd Order System



b. Moore's 2nd Order System



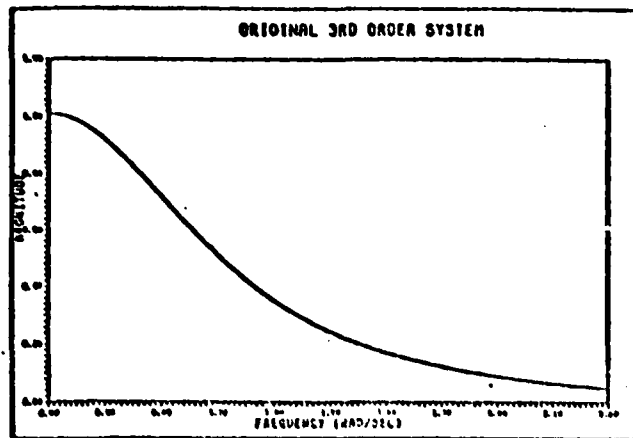
c. El-Attar's 2nd Order System

FIGURE 7. Magnitude Responses for SISO Example

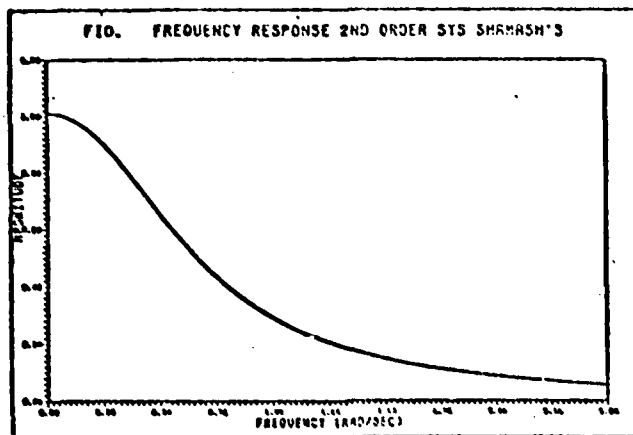
The frequency response of the original third order system is found in Figures 7A, 8A, 9A, 10A. Moore's second order system frequency response (magnitude and phase) is found in Figures 7B, 9B. El-Attar's frequency response for his second order model is located in Figures 7C, 9C. Shamash's system is found in 8B, 10B. Reddy's unstable second order system's frequency response is in Figures 8C, 10C.

For the reduced order transfer functions that are shown in Table III, the Moore algorithm transfer function's dominant roots are the nearest to the origin and have the largest undamped natural frequency of all the algorithms. Yet the time domain and frequency domain responses are very close to those of El Attar and Vidyasagar. It should be noted that the Moore algorithm actually "shifts" its poles and zeros to compensate for the reduction of system order. Shamash's algorithm seems to have performed the best for this particular example. But as the order of the system increases, it becomes increasingly more difficult for a computer to store all of the transfer function coefficients. It is therefore more computationally efficient to work with the state space representation for a large dimension system (Ref 17).

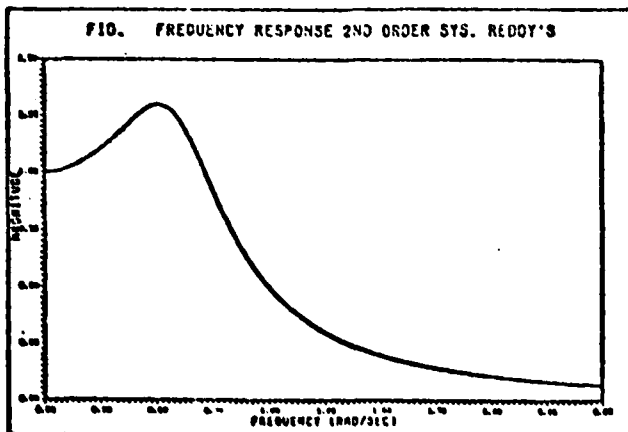
The state space representation of the internally balanced (impulse balanced) system for this example is



a. Original 3rd Order System

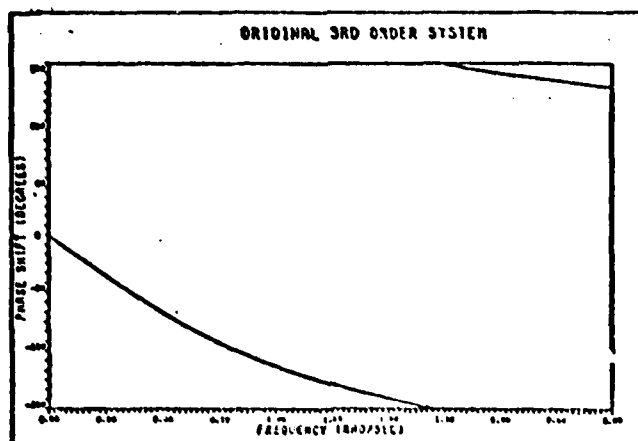


b. Shamash's 2nd Order System

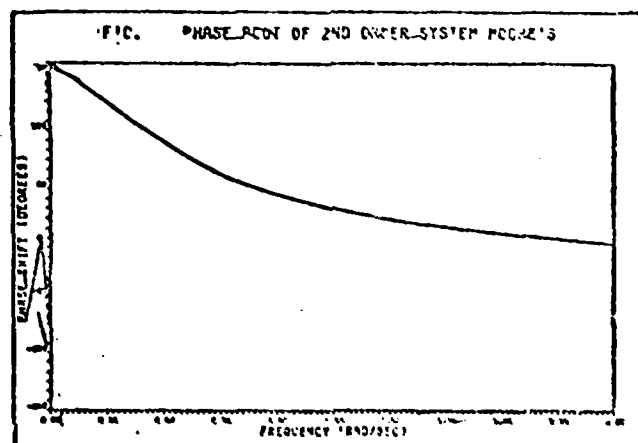


c. Reddy's 2nd Order System

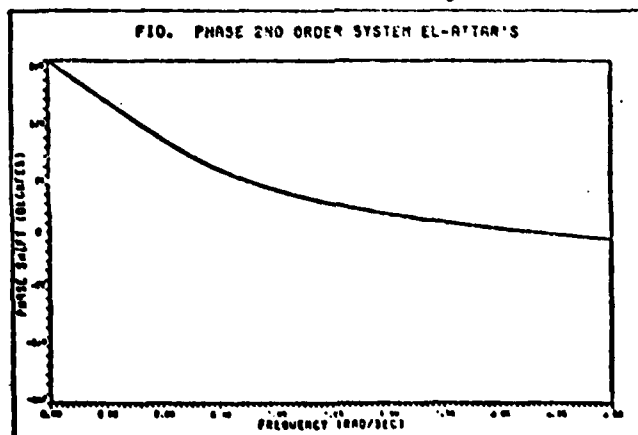
FIGURE 8. Magnitude Responses for SISO Example



a. Original 3rd Order System

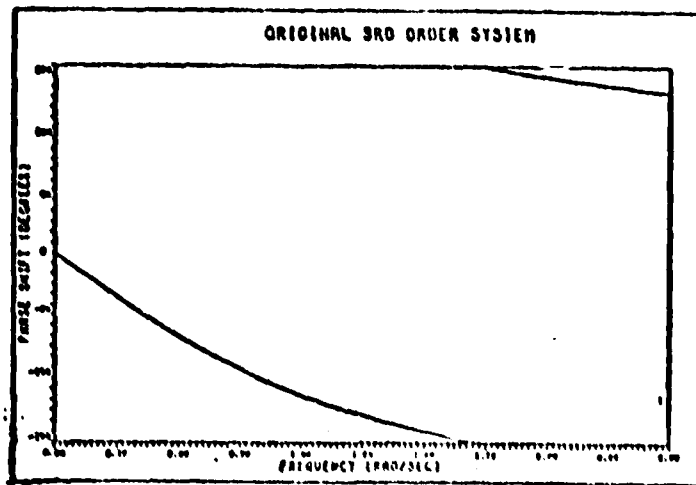


b. Moore's 2nd Order System

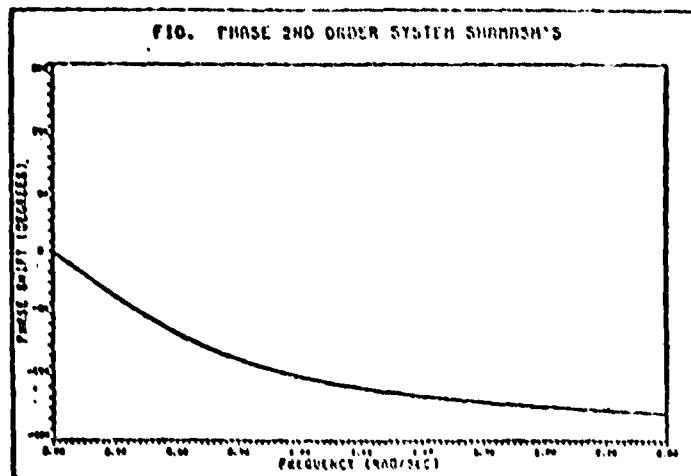


c. El-Attar's 2nd Order System

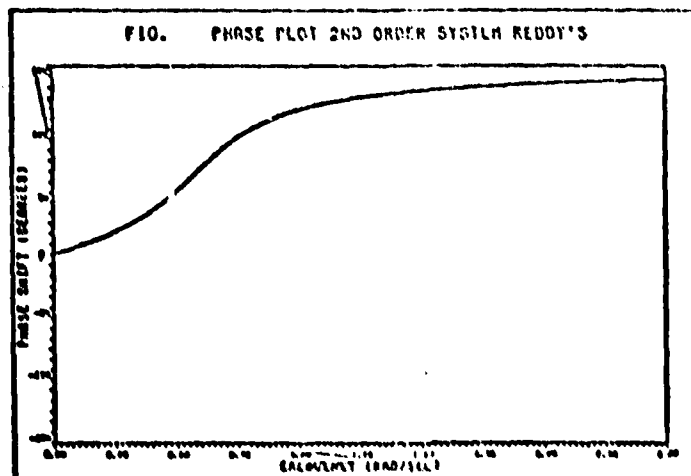
FIGURE 9. Phase Responses for SISO Example



a. Original 3rd Order System



b. Shamash's 2nd Order System



c. Reddy's 2nd Order System

FIGURE 10. Phase Responses for SISO Example

$$A' = \begin{bmatrix} -8.02 \times 10^{-2} & 9.38 \times 10^{-2} & 1.46 \times 10^{-1} \\ -9.38 \times 10^{-2} & -1.95 \times 10^{-2} & 6.17 \times 10^{-2} \\ -1.46 \times 10^{-1} & 6.17 \times 10^{-2} & -1.00 \times 10^0 \end{bmatrix} \quad (65)$$

$$B' = \begin{bmatrix} 7.79 \times 10^{-1} \\ -4.00 \times 10^{-1} \\ 6.68 \times 10^{-1} \end{bmatrix}, \quad C' = \begin{bmatrix} -7.79 \times 10^{-1} & -4.00 \times 10^{-1} & 6.68 \times 10^{-1} \end{bmatrix}$$

Notice that for a SISO system, there is an absolute value symmetry in the balanced A matrix for either step balancing or impulse balancing. In addition, for impulse balancing, as in this case, the balanced B and C matrices will be identical within a change of sign. This property provides a check for computational accuracy. The number of decimal places the symmetrical terms agree to give an indication of the accuracy to which the algorithm converged. This is often useful when the routine within MIMO that solves the two Lyapunov equations for the infinite interval controllability and observability grammians, does not converge satisfactorily. System order is reduced to two by deleting the third state (thus 3rd row of A' , B' , and 3rd column of C').

This low order SISO example was presented to tie together the algorithm and its properties with the tools that actually implement and provide analysis of the algorithm. However, the example is insufficient to illustrate fully the properties of the Moore algorithm.

Therefore the next section shall take a relatively large dimension real world system, apply the Moore algorithm to that system, and analyze and discuss the results of the application.

V. APPLICATION OF MOORE'S ALGORITHM TO B-52E

FLUTTER CONTROL PROBLEM

Introduction

In Section IV, the Moore algorithm was applied to a hypothetical, low order, all pole system. In this section the Moore algorithm is applied to the B-52E flutter control problem, which is a relatively large dimension (24th order), lightly damped, real world system. The airplane is assumed to be in constant altitude, wings-level, steady rectilinear flight (Ref 31).

The attempt here is to compare the input-output properties of the reduced order models obtained via the Moore algorithm with the reduced order models obtained via the AFDDL modified Schwendler and MacNeal technique (Ref 32). The AFDDL technique preserves the physical properties of the original system allowing the resulting reduced order models to be interpreted in a cause-effect (physical relationship) manner. The Moore algorithm, however, preserves only the input-output properties of the original system. Therefore, there will be no attempt to show relationships between the deletion of a state and the physical consequences thereof.

Two measurement devices are available for this test case. They are (see Figure 11): AW 925, which is the vertical wing tip accelerometer, and AW 565, which is the mid-wing vertical

accelerometer. Although the test case allows the use of four inputs (elevator, inboard aileron, outboard flaperon, and outboard aileron) (see Figure 12), only the outboard aileron will be utilized for this test case.

This section is organized as follows. First, the acquisition and modelling of the original system is presented. The "impulse balancing" Moore algorithm is then applied to the system. Subsequently, the original system is input to the "step balancing" Moore algorithm. In each case, results are provided and discussed. Finally, the section is summarized and important points are discussed.

Original System Model Acquisition

The B-52E model, upon which this section is based, is discussed in Reference 31. The dynamic equations for the airplane are in a form for use by Level 3.01.02 FLEXSTAB, which is an analysis tool utilized by AFFDL. Unfortunately, the equations are not in state space form required by the Moore algorithm. Therefore, the derivation of the state space form of the original dynamic equations follows.

Let,

$$\underline{V}_p = ([U \ W \ q]^T)_{3 \times 1} \text{ (airplane rigid body motion degrees of freedom in the body axis system)}$$

$$\underline{\alpha}_{op} = [\theta]_{1 \times 1} \text{ (airplane pitch angle)}$$

$$\underline{U}_1 = ([U_{E1} \ \dots \ U_{E10}]^T)_{10 \times 1} \text{ (in vacuum structural modes)}$$

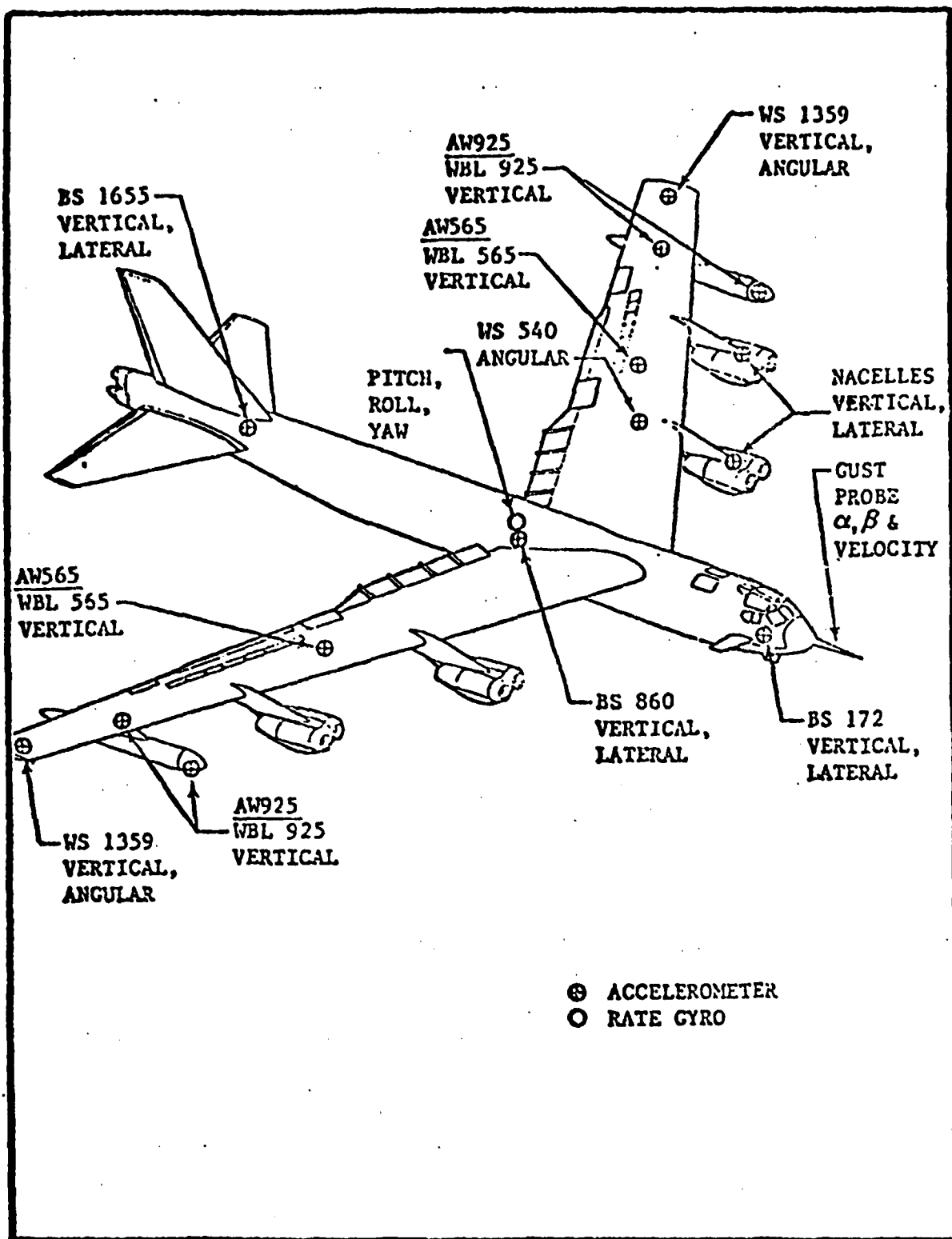


FIGURE 11. Sensor Locations (Ref. 31)

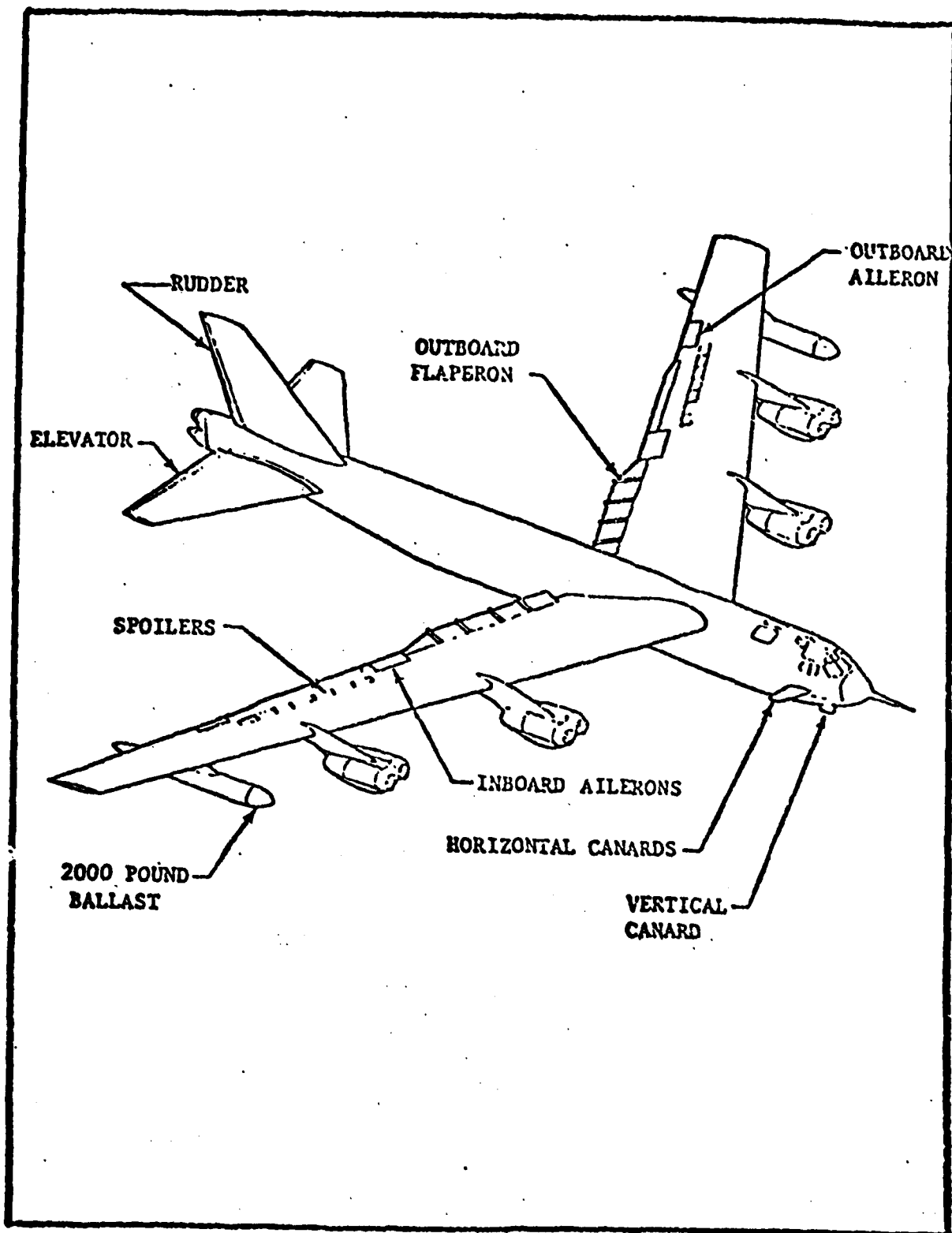


FIGURE 12. B-52E CCV Flight Control Surfaces (Ref. 31)

$$\underline{V}_F = \dot{\underline{U}}_1 \text{ where } \dot{\underline{U}}_1 = I \underline{V}_F$$

$$\underline{T}_{2 \times 1} = \text{vector of measurements}$$

$$\delta c_{4 \times 1} = \text{vector of control inputs}$$

The following equations are the dynamic equations in FLEXSTAB form (ingoring wind gusts) (Ref 31):

$$\begin{aligned} \dot{\underline{V}}_p &= [VP/VP0] \underline{V}_p + [VP/UE0] \underline{U}_1 + (VP/UE1) \dot{\underline{U}}_1 \\ &+ [VP/DEL SO] \delta c + [VP/RO] \underline{r}_{op} \end{aligned} \quad (66)$$

$$\begin{aligned} \ddot{\underline{U}}_1 &= [UE/VP0] \underline{V}_p + [UE/UE0] \underline{U}_1 + [UE/UE1] \dot{\underline{U}}_1 \\ &+ [UE/DEL SO] \delta c \end{aligned} \quad (67)$$

$$\dot{\underline{r}}_p = [RP/VP0] \underline{V}_p + [RP/RP] \underline{r}_{op}$$

The measurement equation is:

$$\begin{aligned} \underline{T} &= [T/VP0] \underline{V}_p + [T/UE0] \underline{U}_1 + [T/UE1] \dot{\underline{U}}_1 \\ &+ [T/UE2] \ddot{\underline{U}}_1 + [T/DEL SO] \delta c + [T/VP1] \dot{\underline{V}}_p \\ &+ [T/RO] \underline{r}_p + [T/RI] \dot{\underline{r}}_{op} \end{aligned} \quad (68)$$

The measurement equation (Equation 68) has a second derivative term in it. Therefore, by letting $\dot{\underline{U}}_1 = I \underline{V}_f$,

$$\dot{\underline{U}}_1 = I \underline{V}_f \quad (69)$$

and augmenting this extra state onto the state vector the problem of the second derivative is reduced to two first order derivatives. The actual state space form of the equations is now presented.

In the state space form, $\dot{\underline{x}} = \underline{A}\underline{x} + \underline{B}\underline{u}$, the dynamic equations become

$$\begin{Bmatrix} \dot{\underline{v}}_p \\ \dot{\underline{r}}_{op} \\ \dot{\underline{v}}_f \\ \dot{\underline{u}}_1 \end{Bmatrix}_{24 \times 1} = \begin{bmatrix} V_p/V_{p0} & V_p/R_o & V_p/U_{E1} & V_p/U_{E0} \\ R_p/V_{p0} & R_p/R_o & 0 & 0 \\ U_E/V_{p0} & 0 & U_E/U_{E1} & U_E/U_{E0} \\ 0 & 0 & I & 0 \end{bmatrix}_{24 \times 24} \begin{Bmatrix} \underline{v}_p \\ \underline{r}_{op} \\ \underline{v}_f \\ \underline{u}_1 \end{Bmatrix}_{24 \times 1} + \begin{bmatrix} V_p/DEL SO \\ 0 \\ U_E/DEL SO \\ 0 \end{bmatrix}_{24 \times 4} \delta_{c4 \times 1} \quad (70)$$

The measurement equation is:

$$\begin{aligned} [\underline{T}]_{2 \times 1} &= [T/V_{p0} \quad T/R_o \quad T/U_{E1} \quad T/U_{E0}]_{2 \times 24} \begin{bmatrix} \underline{v}_p \\ \underline{r}_{op} \\ \underline{v}_f \\ \underline{u}_1 \end{bmatrix} + \\ &+ [T/V_{p1} \quad T/R_{p1} \quad T/U_{E2} \quad 0]_{2 \times 24} \begin{bmatrix} \dot{\underline{v}}_p \\ \dot{\underline{r}}_{op} \\ \dot{\underline{v}}_f \\ \dot{\underline{u}}_1 \end{bmatrix} + [T/DEL SO]_{2 \times 24} \delta_c \end{aligned} \quad (71)$$

Substituting in for the first order term in equation (71), yields the following measurement equation:

$$[\underline{T}] = ([T/V_{p0} \ T/R_0 \ T/U_{E1} \ T/U_{E0}] + [T/V_{p1} \ T/R_{p1} \ T/U_{E2} \ 0]) \underline{x} \\ + ([T/V_{p1} \ T/R_{p1} \ T/U_{E2} \ 0] [B] + [T/DEL SO]) \underline{\delta}_c \quad (72)$$

A computer program was written that formed the A, B, C, and D matrices. Using data supplied by AFFDL, the system was obtained. The A matrix eigenvalues are shown in Table IV.

TABLE IV

The Eigenvalues for the B-52 Test Case

<u>Real Part</u>	<u>± Imaginary Part</u>	<u>Period (sec)</u>
-.3971	.3587E+02	.1752
-.5124	.3603E+02	.1745
-.4651	.2740E+02	.2294
-.1918E+01	.2421E+02	.2604
-.6338	.1996E+02	.3150
-.1445E+01	.1923E+01	.4956
-.9345E-03	.6578E-01	.9556E+02
-.6173	.6154E+01	.1027E+01
-.2137E+01	.1607E+02	.3947
-.05404	.1551E+02	.4051
-.1142	.1238E+02	.5077
-.1464	.1254E+02	.5012

A computer program was written to interface with TOTAL (Ref 21) to obtain time domain and frequency domain responses. The time responses of the full twenty-fourth order model to a .1 radian step input were obtained. Figure 13 shows the comparison of AW925 (in/sec²) for AFFDL's FLEXSTAB case, and the state space model in response to .1 radian step input. Figure 14 shows the comparison for AW565 (in/sec²). This verifies that the correct model has been attained. The Moore algorithm is now applied to this state space system.

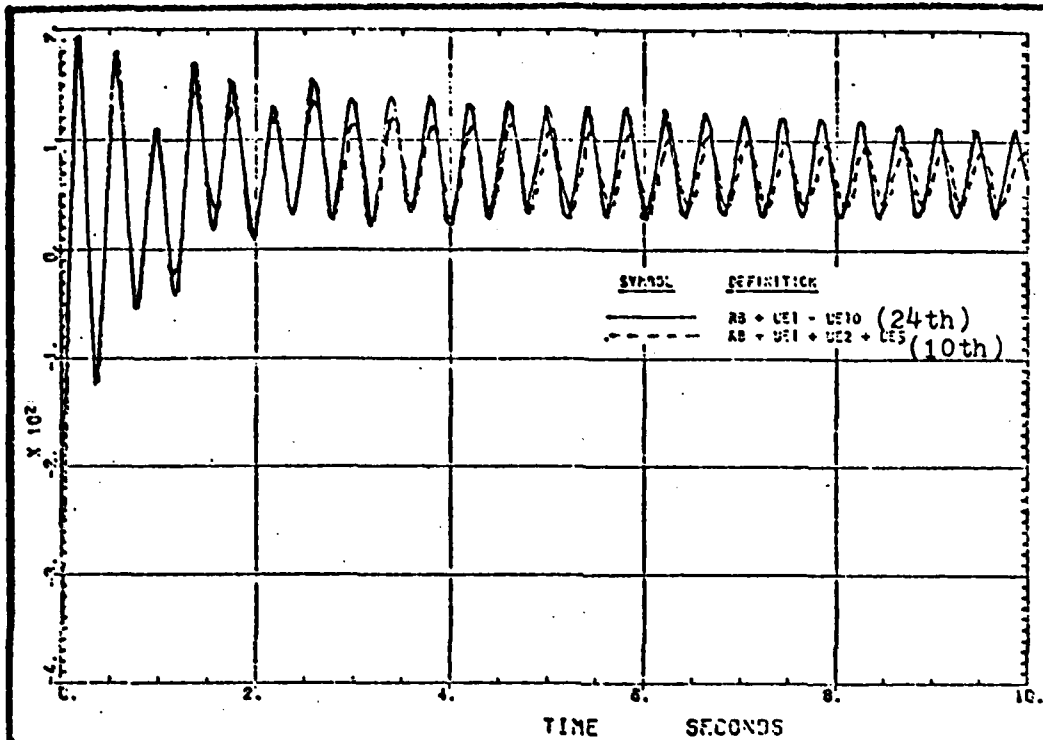
Application of Moore "Impulse Balancing" Algorithm

The Moore "impulse balancing" algorithm is now applied to the B-52E flutter control problem state space model. To assess the performance of the resulting reduced order model, both the step time response and the frequency response will be compared to the full 24th order system "truth model".

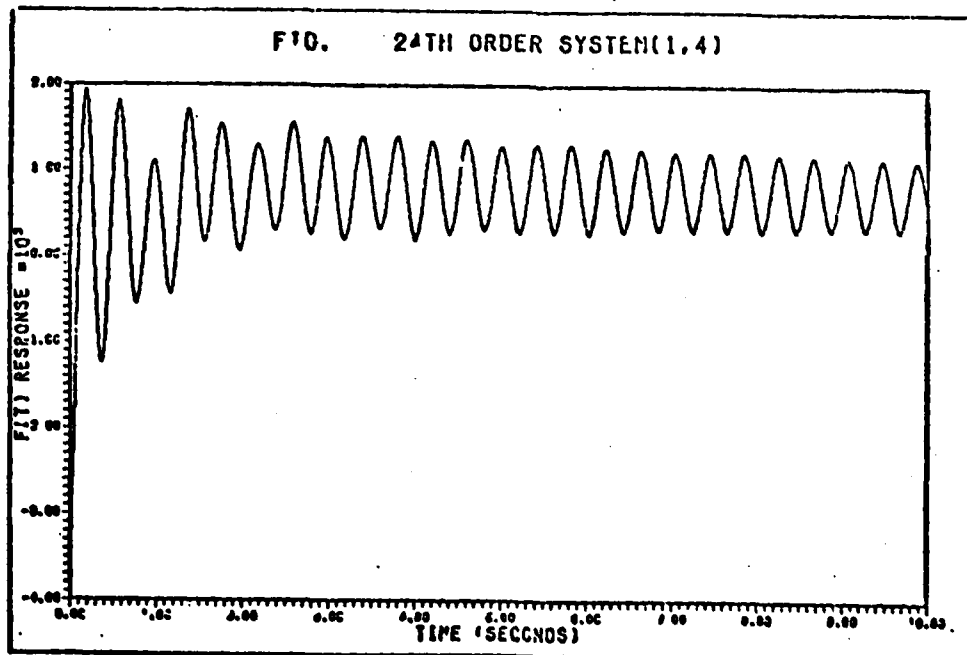
Because we examine the time response performance with a step input an important point should be noted. In Section II, theory was presented that claimed if the algorithm is designed for the particular input that the reduced order models will be evaluated against, then lower order models can probably be attained for similar levels of reproduction accuracy. By subjecting the B-52E flutter control problem reduced order models to a step input, for either the impulse balanced reduced

FIGURE 7. Magnitude Responses for SISO Example

52

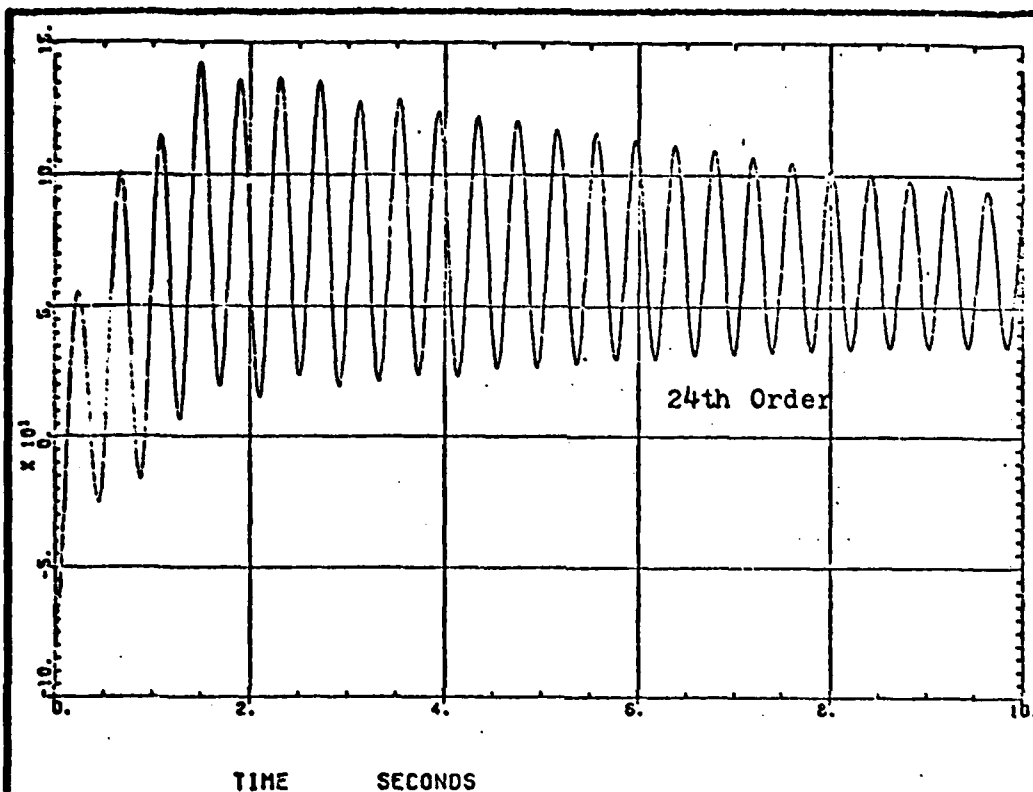


a. AW925 (in/sec²) Response To .1 Radian Step (FLEXSTAB)

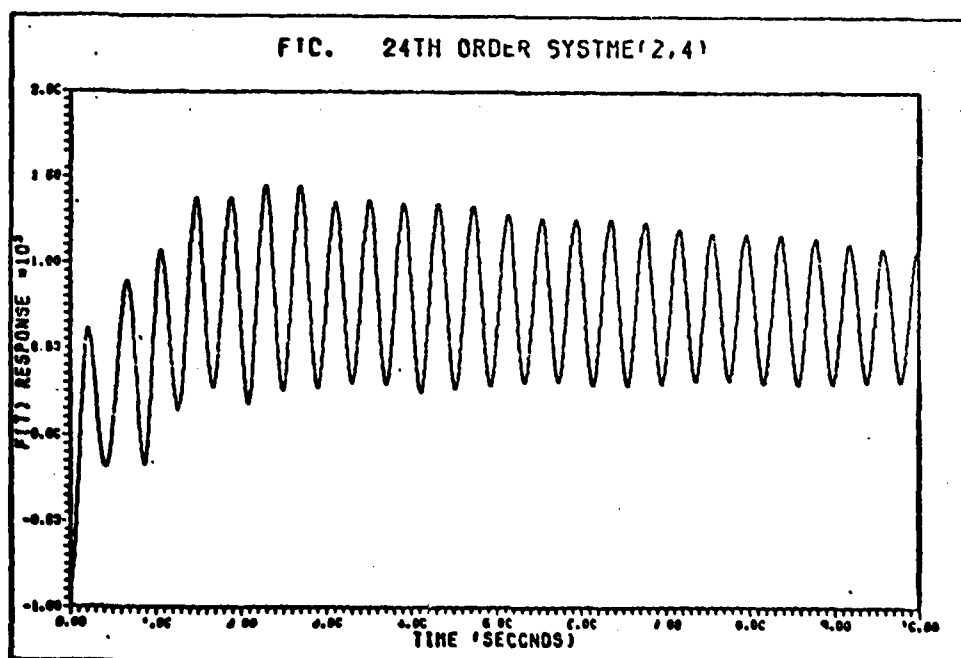


b. AW925 (in/sec²) Response To .1 Radian Step (State Space)

FIGURE 13. Response To .1 Radian Step in Outboard Aileron



a. AW565 (in/sec²) Response to .1 Radian Step (FLEXSTAB)



b. AW565 (in/sec²) Response To .1 Radian Step (STATE SPACE)

FIGURE 14. Response To .1 Radian Step in Outboard Aileron

order model, or the step balanced reduced order model, we would expect that the step balancing algorithm should give better time response performance results.

Results. Figure 15 contains the response of AW925 (vertical wing-tip accelerometer) to a .1 radian step in outboard aileron for the full order original system, and the twentieth, eighteenth, and tenth order impulse balanced reduced order models respectively. A definite degradation in both transient response and steady state response occurs as the system order is reduced. Similar results are pictured in Figure 16 for AW565 (vertical mid-wing accelerometer) for a .1 radian step in outboard aileron.

In Figure 17, the frequency response for AW925/OBA is shown for the twenty-fourth (full order), twentieth, eighteenth, and tenth order systems. As the system order is reduced, essential high frequency information is being lost. However, the low frequency portion of the responses seems to be relatively constant. Figure 18 shows similar results for AW565/OBA.

Figure 19 and 20 show the phase responses of AW925/OBA and AW565/OBA, respectively. As in Figures 17 and 18 the high frequency information is being discarded. This suggests that sinusoidal inputs at these higher frequencies to the reduced order systems would not produce accurate replication of the full-order system.

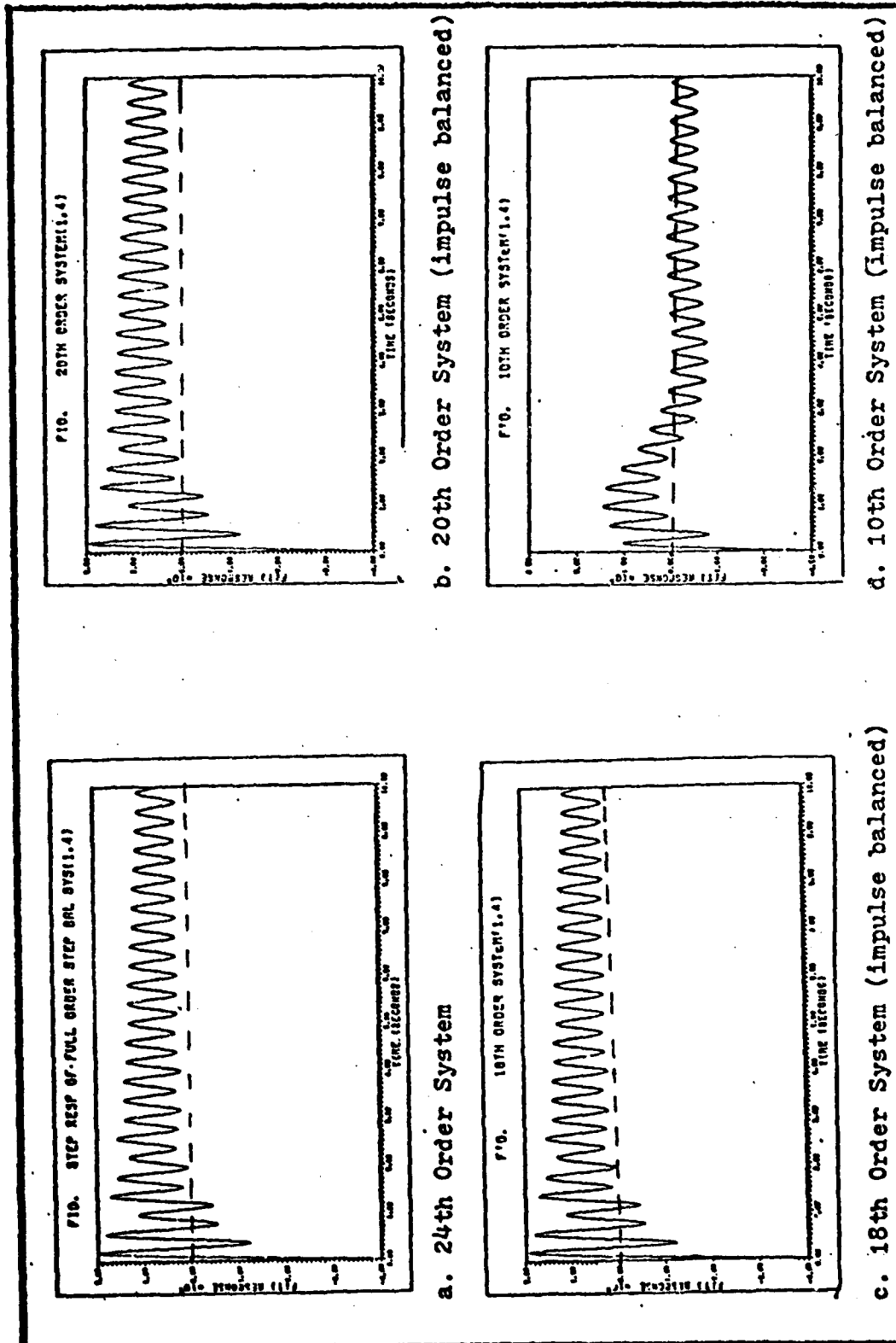
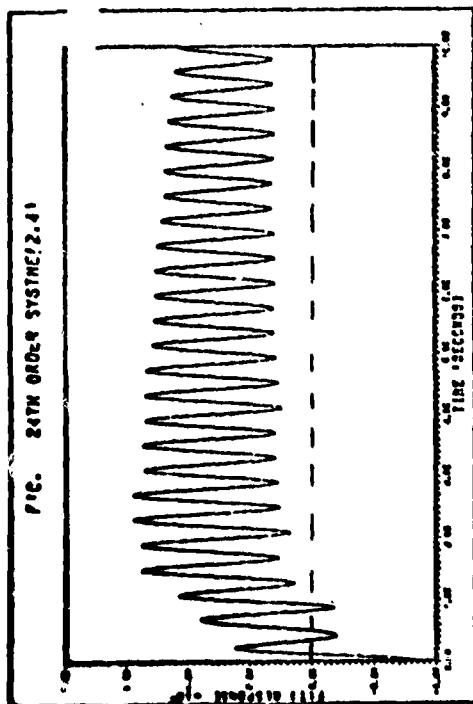
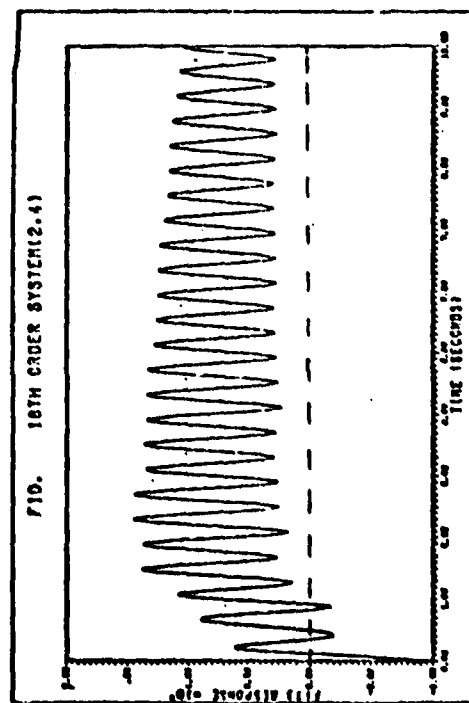


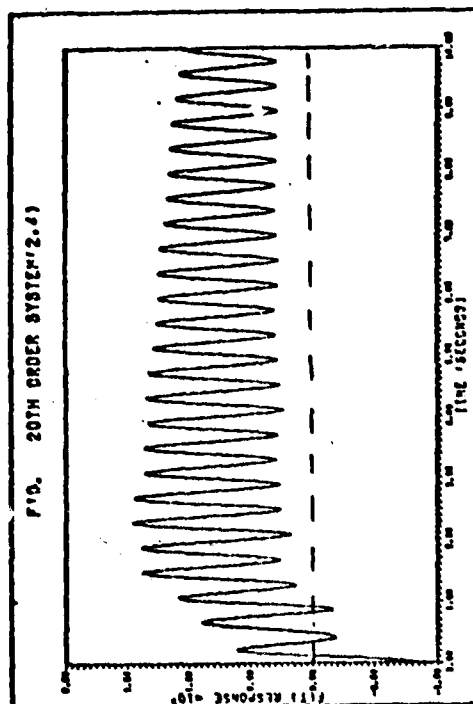
FIGURE 15. AW925/OBA To .1 Rad. Step Input
Using Impulse Balancing



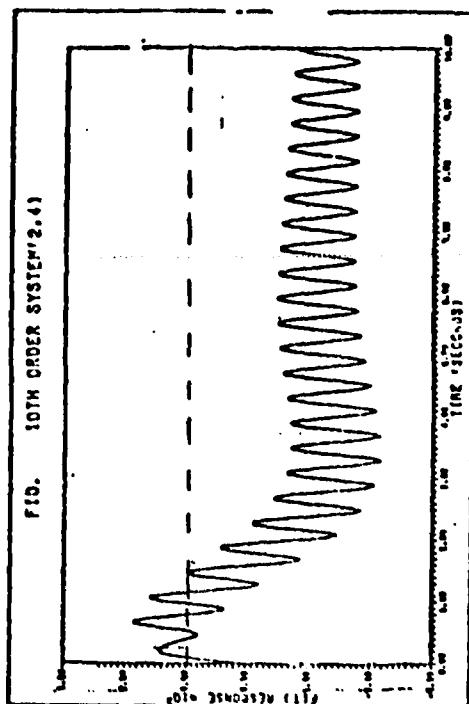
a. 24th Order System (Full Order)



c. 18th Order System (impulse balanced)

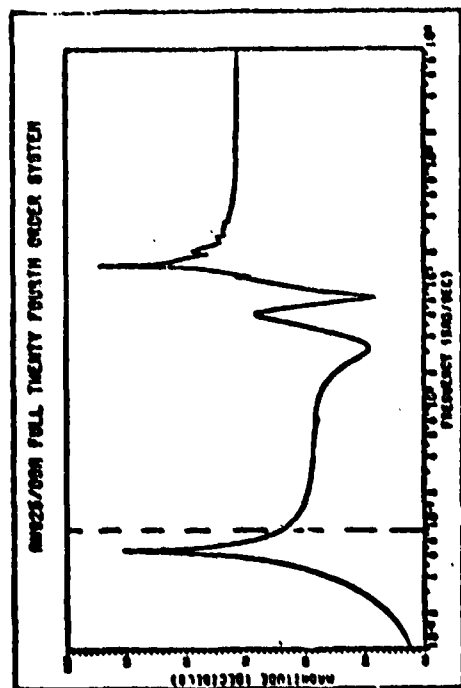


b. 20th Order System (impulse balanced)

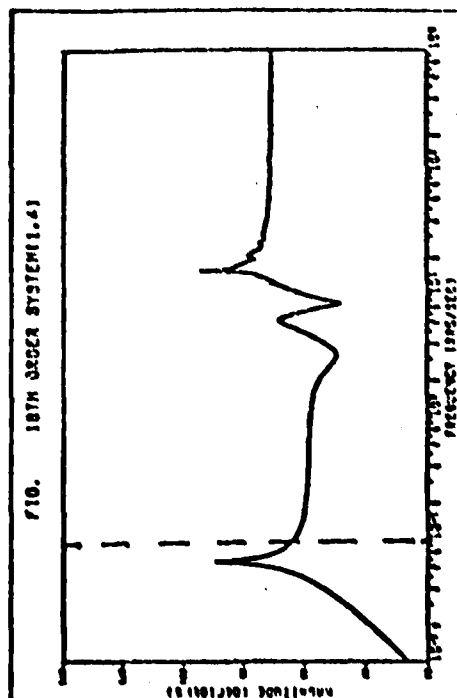


d. 10th Order System (impulse balanced)

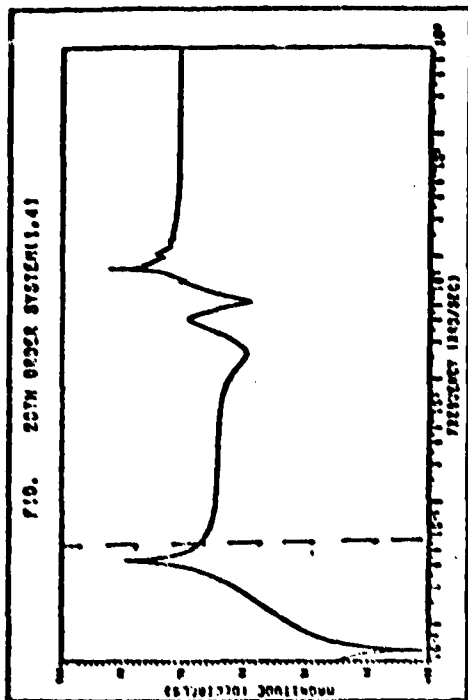
FIGURE 16. AW565/OBA To .1 Rad. Step Input
Using Impulse balancing



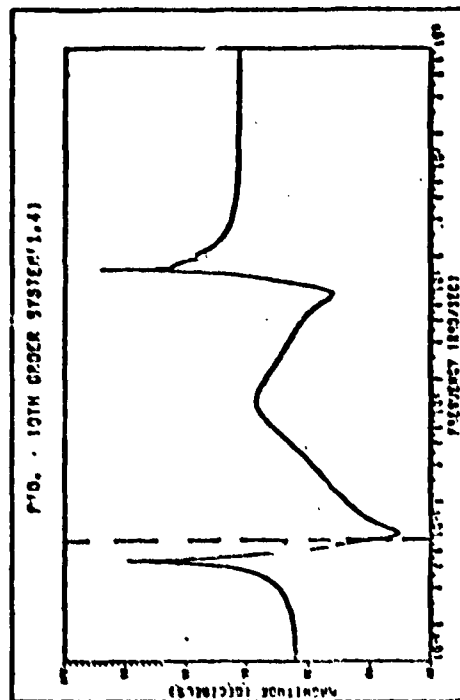
a. 24th Order System (Full Order)



c. 18th Order System (impulse balanced)

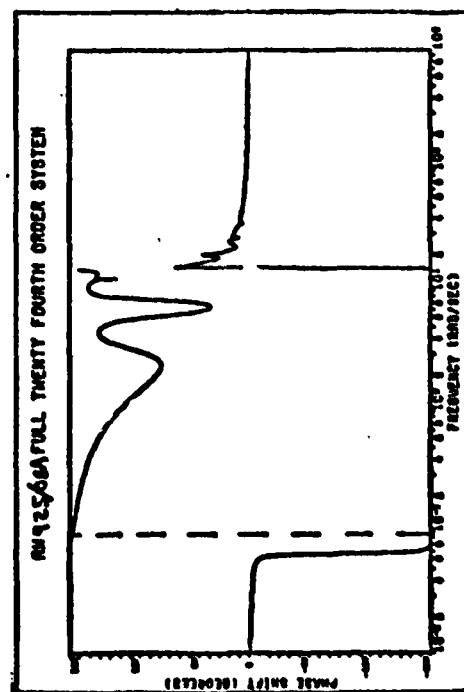


b. 20th Order System (impulse balanced)

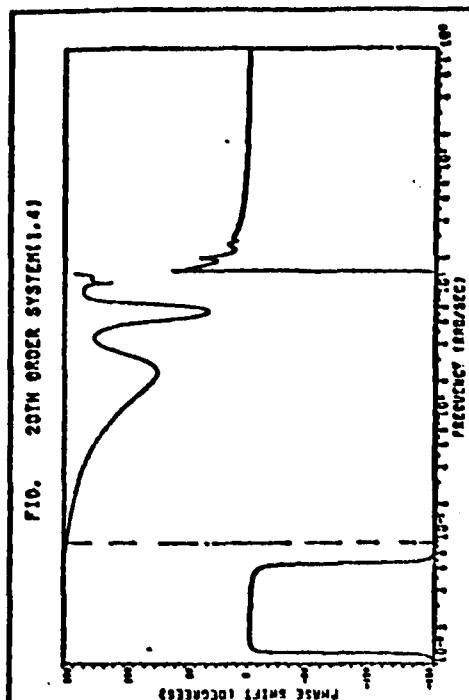


d. 10th Order System (impulse balanced)

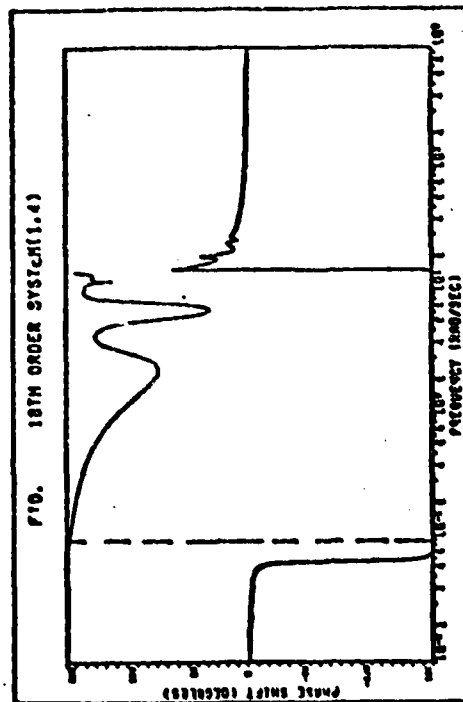
FIGURE 17. AW925/00A Frequency Response
Using Impulse Balancing



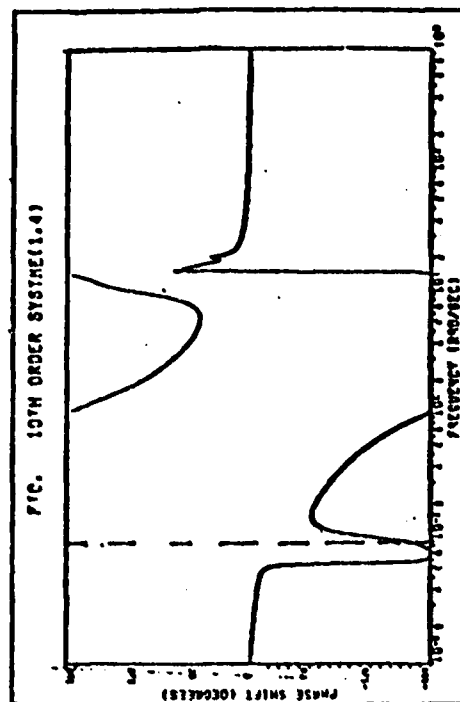
a. 24th Order System (Full Order)



b. 20th Order System (impulse balanced)

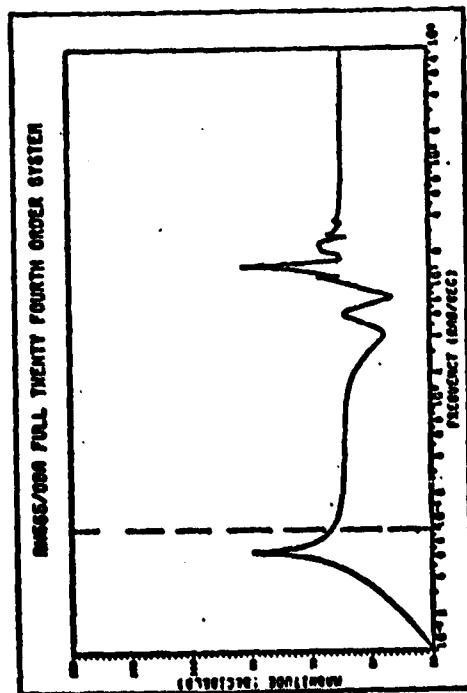


c. 18th Order System (impulse balanced)

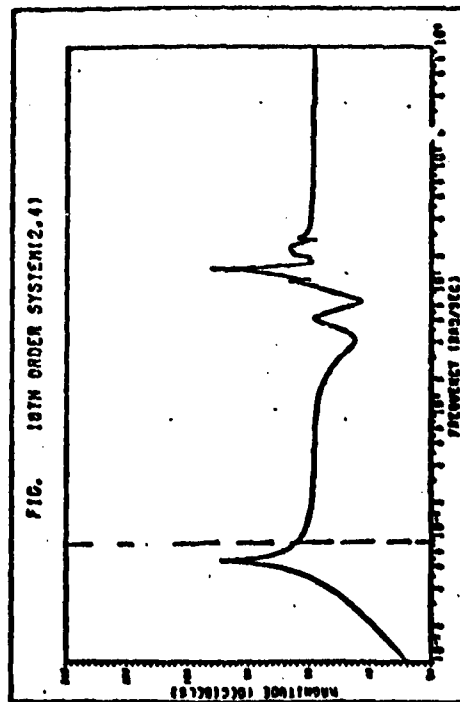


d. 10th Order System (impulse balanced)

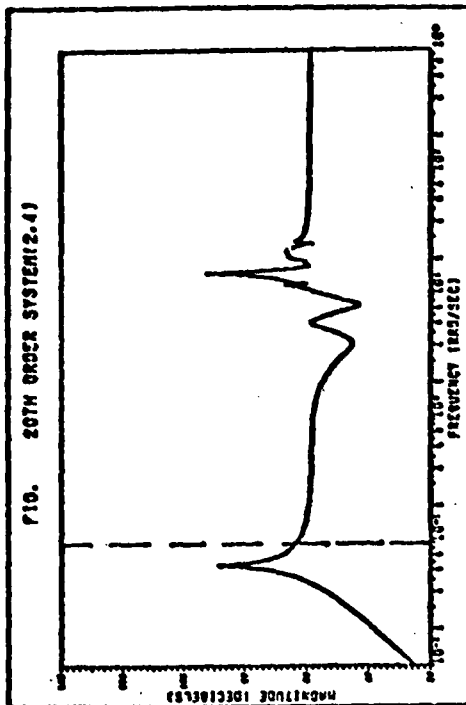
FIGURE 18. AW925/OBA Phase Response
Using Impulse Balancing



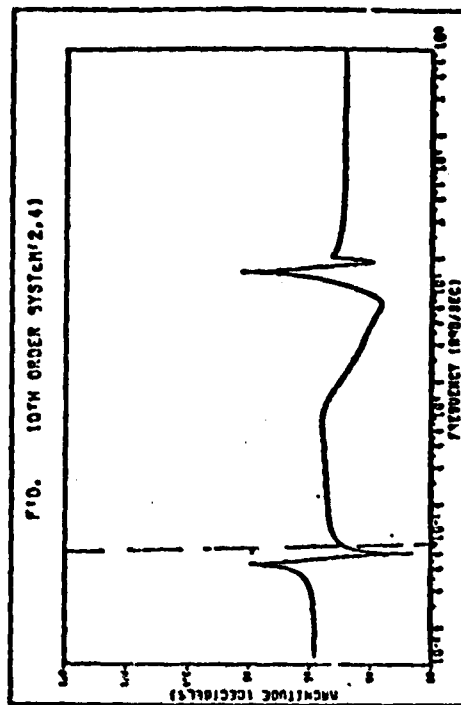
a. 24th Order System (Full Order)



c. 18th Order System (impulse balancing)

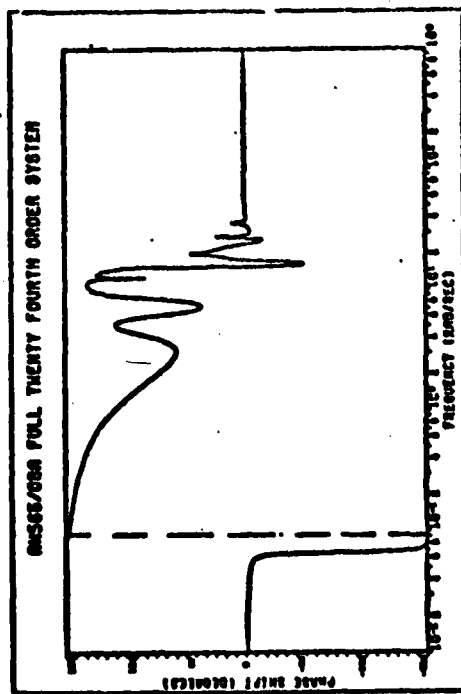


b. 20th Order System (impulse balancing)

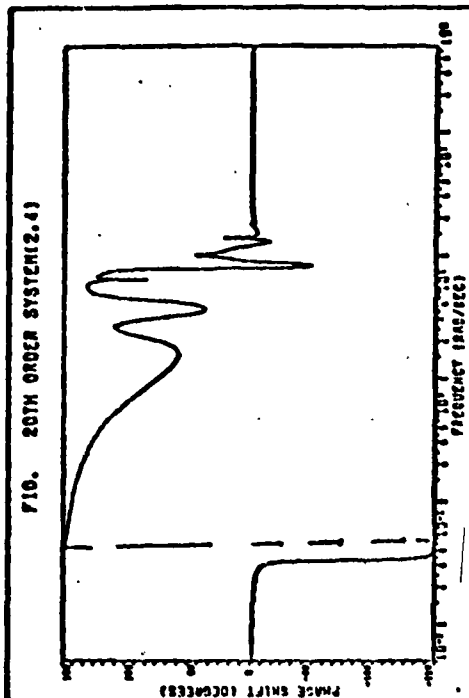


d. 10th Order System (impulse balancing)

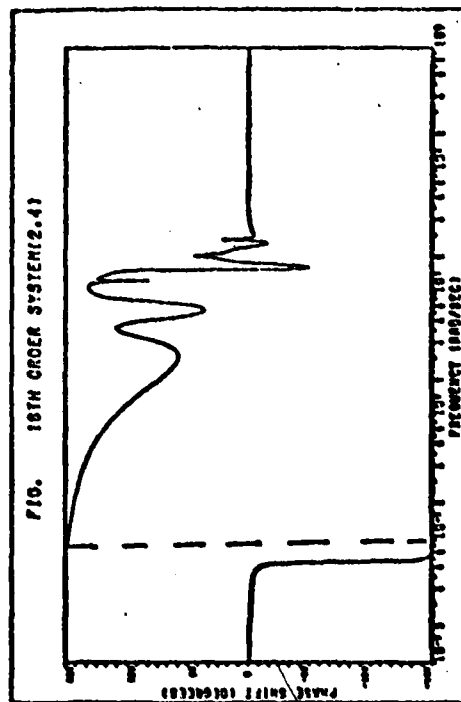
FIGURE 19. AW565/OBA Frequency Response
Using Impulse Balancing



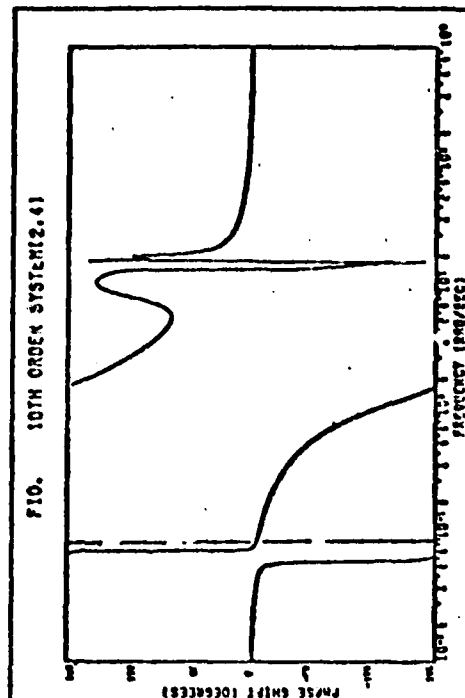
a. 24th Order System (Full Order)



b. 20th Order System (Impulse balanced)



c. 18th Order System (Impulse balanced)



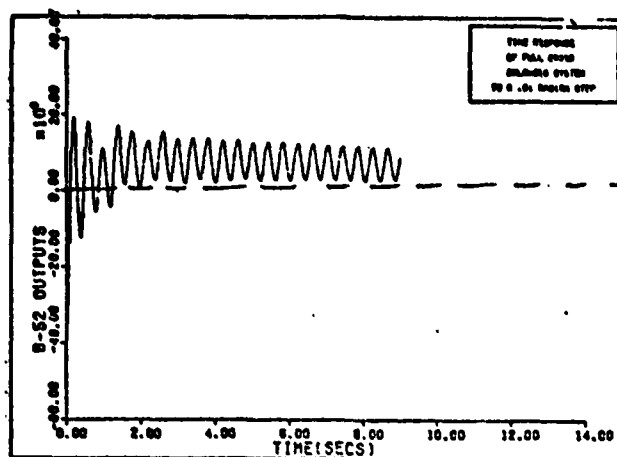
d. 10th Order System (Impulse balanced)

FIGURE 20. AW565/OBA Phase Response
Using Impulse Balancing

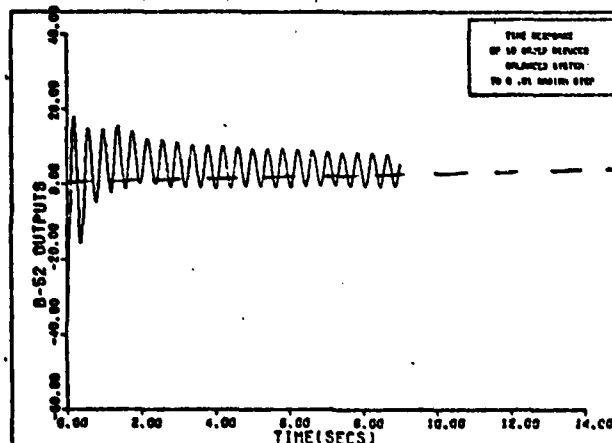
Figures 21 and 22 show the responses of AW925 and AW565 to a .1 radian step in outboard aileron respectively. These plots show the full twenty-fourth order, the fourteenth order, and the thirteenth order systems for both Figures 21 and 22. There is a slight steady state offset occurring in 21b and 22b (fourteenth order). However, severe transient and steady state errors exist in the thirteenth order models in Figures 21c and 22c. This suggests that fourteenth order is the lowest order this system can be reduced to when impulse balancing is applied but a step is the actual input used for evaluation.

H_{INF} Matrix Investigation. As mentioned in Section II, Appendix B, and References 24, 25, there is a byproduct of this algorithm which gives an estimate of the lowest possible order the system can be reduced to without severe loss of reproduction accuracy (subjective). This byproduct is referred to by Moore in References 24 and 25 as the "H_{INF}" matrix.

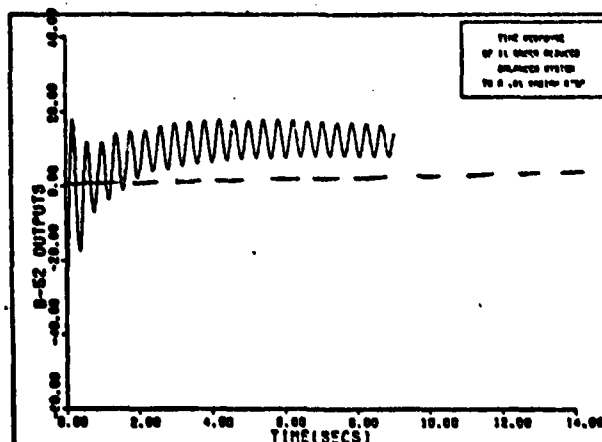
In theory, the singular values of the H_{INF} matrix provide a guideline to the choice of the lowest possible order the system can be reduced for accurate original system replication. However, this test case uses a step input to an impulse balanced system. Therefore, the results may not be as accurate as later in this section when the system is subjected to step balancing.



a. 24th Order System

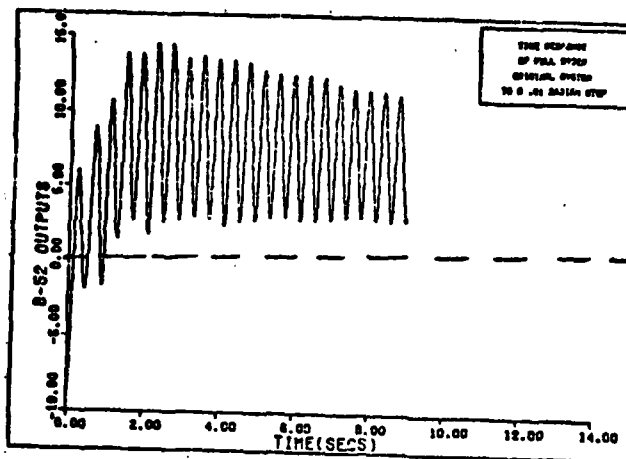


b. 14th Order System

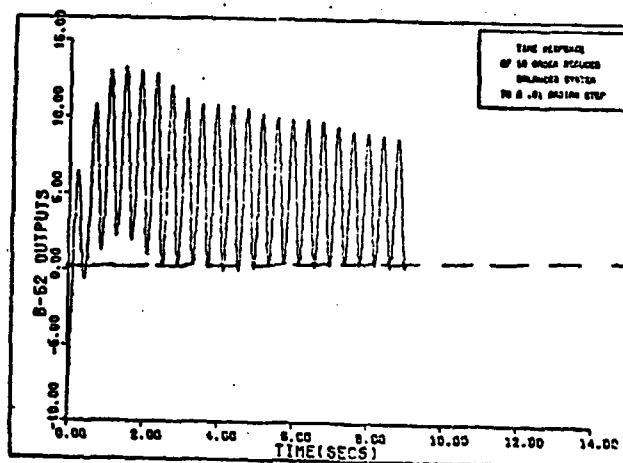


c. 13th Order System

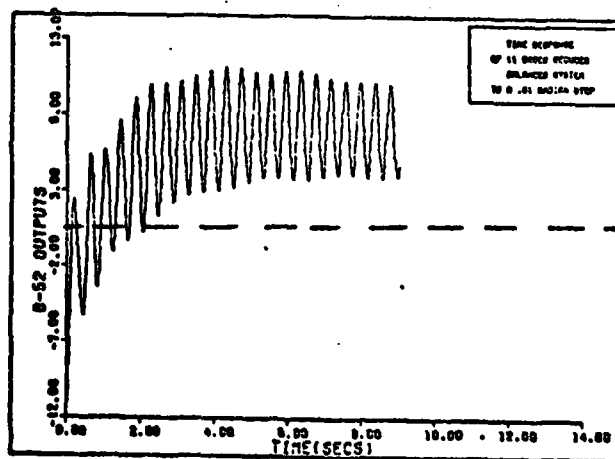
FIGURE 21. AW925/OBA to .1 Radian Step Input
Using Impulse Balancing



a. 24th Order System



b. 14th Order System



c. 13th Order System

FIGURE 22. AW565/OBA to .1 Radian Step Input
Using Impulse Balancing

The singular values of the H_{INF} matrix for impulse balancing are shown in Table V. The singular values are listed in descending order. As discussed in Section II the number of "large" singular values that cluster together provide a guideline for the choice of minimum dimension of the reduced order model. There is no clear dividing line. However, one of the largest ratios is between the fifth and sixth singular values. This indicates that for impulse balancing coupled with an impulse input, the system order could possibly be reduced to fifth order without severe loss of accuracy. Unfortunately, this system has a D matrix (feedforward) and an impulse input cannot be applied.

TABLE V.

Singular Values of the H_{INF} Matrix for Impulse Balancing

2.2443E+05	8.8185E+03	1.8684E+03	3.3900E+02
2.1825E+05	6.0901E+03	2.1874E+03	3.3819E+02
1.3755E+05	5.2009E+03	2.1445E+03	2.9344E+02
1.3723E+05	3.6827E+03	2.0217E+03	2.8774E+02
1.2722E+05	3.5674E+03	1.4901E+03	9.3683E+01
1.2131E+04	3.1964E+03	1.3496E+03	9.2485E+01

The fourteenth order impulse balanced system produces a relatively good reproduction of the original system in response to a .1 radian step in outboard aileron. However, if the theory holds, a definite improvement (lower attainable system order) should be achieved through the use of step balancing.

Application of Moore "Step Balancing" Algorithm

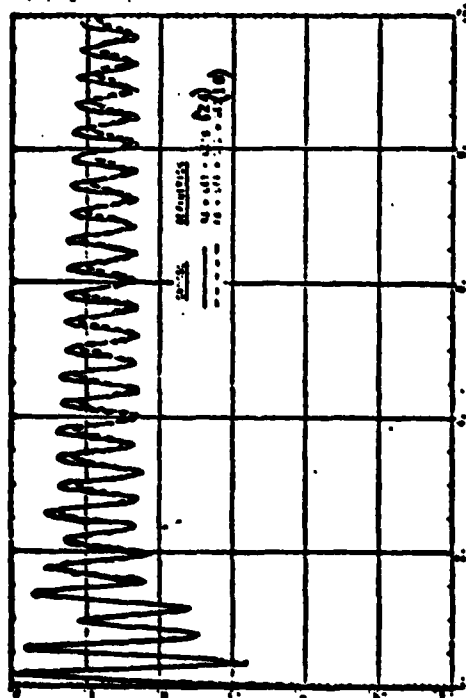
By utilizing the two modifications to the impulse balancing algorithm described in Section II, specifically:

- 1) input $(A, A^{-1}B, C)$ (provided A^{-1} exists) to the impulse balancing algorithm

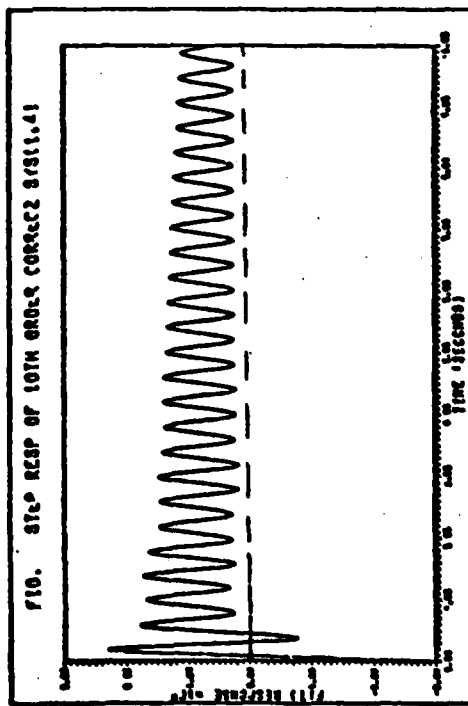
- 2) add a D matrix to compensate for steady state offset

the step balancing algorithm is realized. The results will show that when the algorithm is "optimized" for a given input, in this case a step input, lower order models are achievable for a given level of system reproduction accuracy.

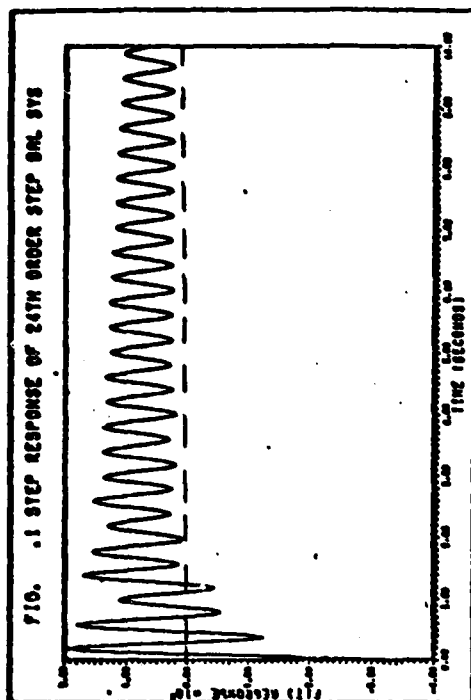
Results. In Figure 23, AW925 to a .1 radian step in outboard aileron is shown for the full twenty fourth order (AFFDL's and this thesis'), the tenth order (AFFDL's and this thesis'), and the ninth order (this thesis') models. In Figure 23a AFFDL's twenty fourth order and tenth order systems are shown. The solid curve is the twenty fourth order system; the dashed curve, the tenth order model. Notice that the tenth order response has a definite phase shift as time increases. It also seems to be damped slightly more than the original system. However, it gives an excellent reproduction of the transient



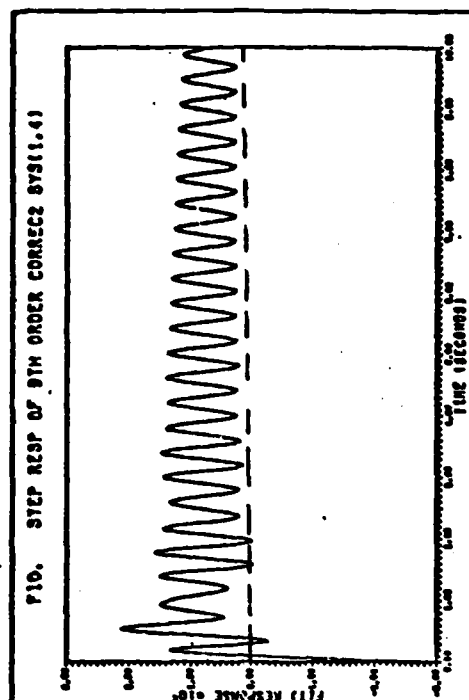
a. APFDL'S 24th and 10th Order Systems (FLEXSTAB)



c. 10th Order System (step balancing)



b. Full Order System (state-space)

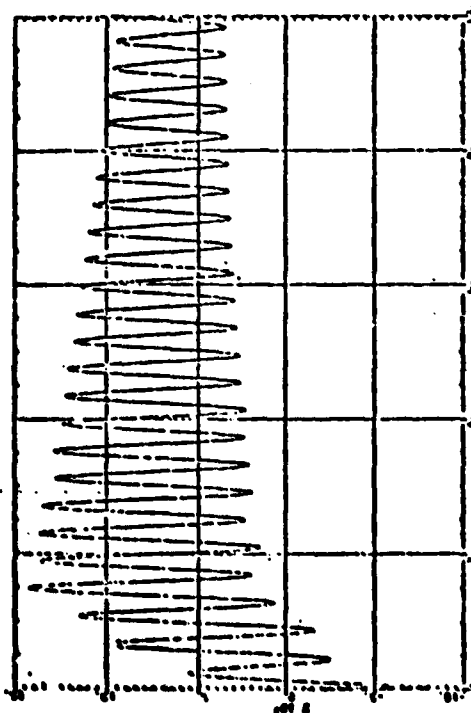


d. 9th Order System (step balancing)

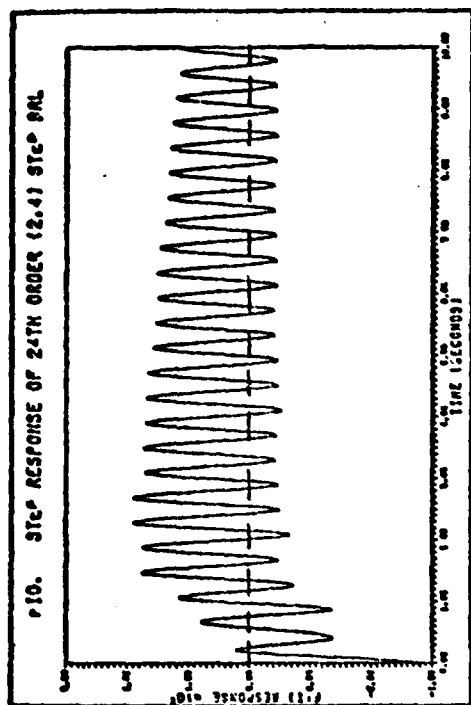
FIGURE 23. AW925/OBA to .1 Rad. Step Input Using Step Balancing

response and the frequency of oscillation. Figures 23c and 23d show the tenth and ninth order models obtained via the Moore step balancing algorithm. The transient reproduction ability is low for both of these systems. However, there is no phase shift or excessive damping as found in the AFFDL model. Figure 24 shows similar results for AW565 to a .1 radian step in outboard aileron. Table VI shows the percent differences in magnitude between the twenty fourth order system and the tenth and ninth order systems for both AW925 and AW565 as a function of time. The table illustrates the large magnitude of transient errors and the small magnitude of steady state errors.

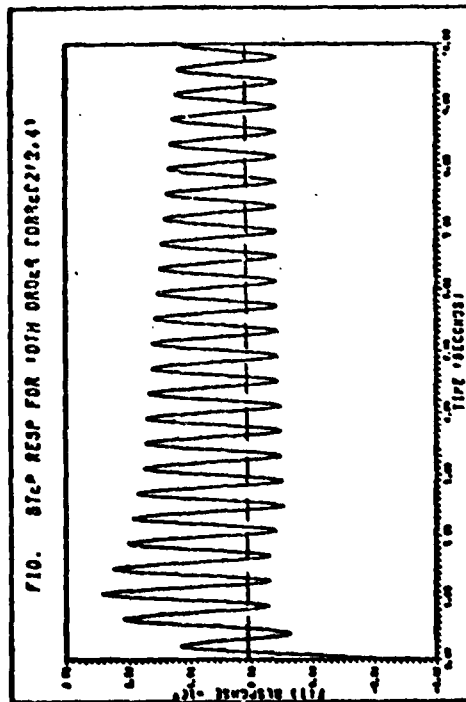
The frequency responses of the twenty fourth, tenth, and ninth order systems for AW925/OBA and AW565/OBA are shown in Figures 25 and 26, respectively. Figures 27 and 28 show the phase responses for AW925/OBA and AW565/OBA, respectively. Important high frequency, magnitude and phase information is being lost as the system order is reduced. However, comparing Figures 25 with 17 for impulse balancing, 26 with 19 for impulse balancing, 27 with 19 for impulse balancing, and 28 with 20, the amount of high frequency information retained by the step balanced reduced order models shows a definite improvement over the impulse balanced reduced order model. Also the tenth order time response for step balancing (Figures 23c, 24c) are



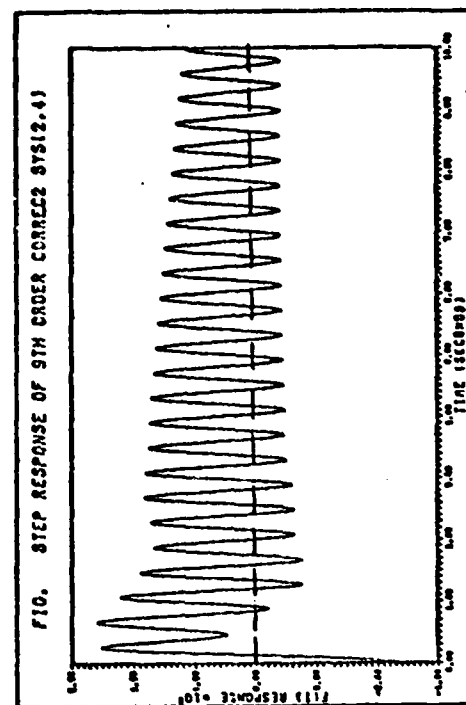
a. APPDL'S 24th Order System
(PLEXSTAB)



b. Full Order System (state-space)



c. 10th Order System (step balancing)



d. 9th Order System (step balancing)

FIGURE 24. AW565/OBA to .1 Rad. Step Input
Using Step Balancing

TABLE VI

Differences in Time Response Between 24th and 10th and 9th
Order Systems

<u>T</u>	<u>DIF. AW925</u> <u>24-10(%)</u>	<u>DIF.</u> <u>24-9(%)</u>	<u>DIF. AW565</u> <u>24-10(%)</u>	<u>DIF.</u> <u>24-9(%)</u>
0	0	0	0	0
.1	69.58	318.61	214.51	87.23
.2	16.52	35.34	43.98	58.71
.3	254.58	101.17	64.81	83.7
.4	64.65	123.83	179.32	117.95
.5	2.07	47.66	95.25	97.25
.6	4.11	1.79	50.53	56.62
.7	81.05	73.31	40.98	89.33
.8	157.66	160.33	71.16	79.83
.9	47.9	59.89	123.72	125.37
1.0	35.91	20.16	44.42	35.37
1.2	197.26	218.0	48.48	41.31
1.4	4.48	8.69	23.77	5.43
1.6	1.16	104.76	17.96	15.80
1.8	8.95	7.81	10.22	6.16
2.0	61.34	92.13	15.74	6.04
2.2	8.97	4.49	2.79	10.59
2.4	6.98	29.21	1.71	9.32
2.6	6.76	16.24	3.00	11.633
2.8	26.85	25.77	5.26	6.67
3.0	7.84	7.21	4.64	3.73
4.0	11.02	20.8	3.19	4.92
5.0	5.61	7.28	3.77	5.32
6.0	1.64	6.48	2.42	2.99
7.0	5.04	5.21	4.54	4.41
8.0	2.23	7.52	2.13	2.7
9.0	5.94	3.56	7.2	6.7
10.0	7.17	3.42	4.72	4.06

AD-A080 371

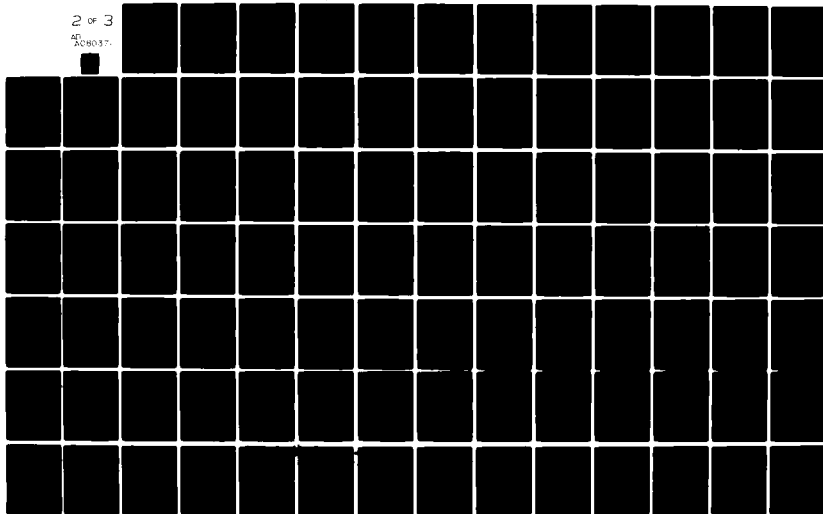
AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/6 20/4
MODEL ORDER REDUCTION USING THE BALANCED STATE REPRESENTATION! --ETC(U)
DEC 79 J R MCCLENDON
AFIT/6E/EE/79-22

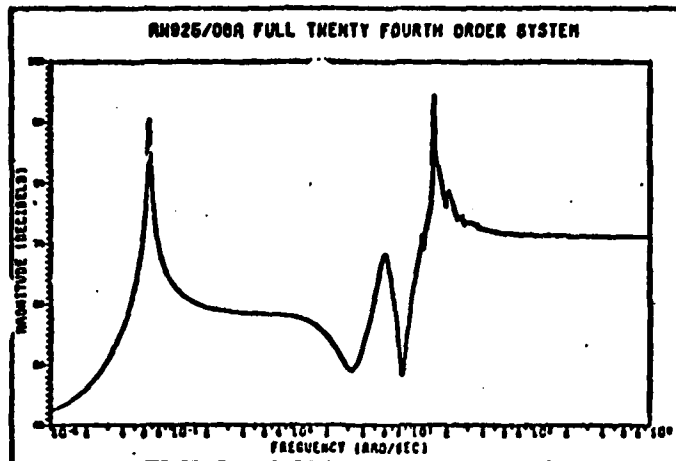
UNCLASSIFIED

NL

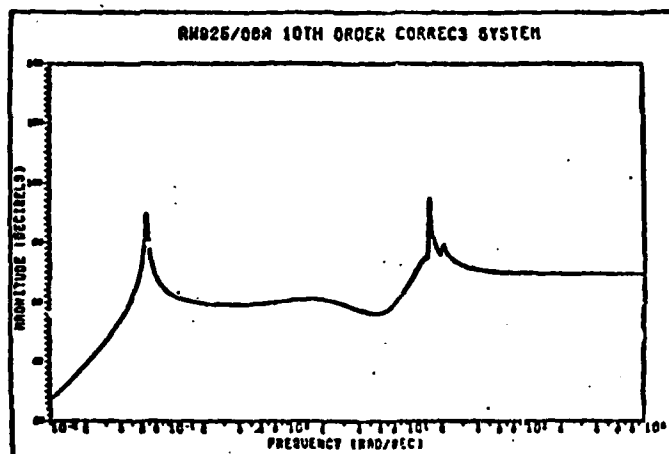
2 of 3

41
AC8057

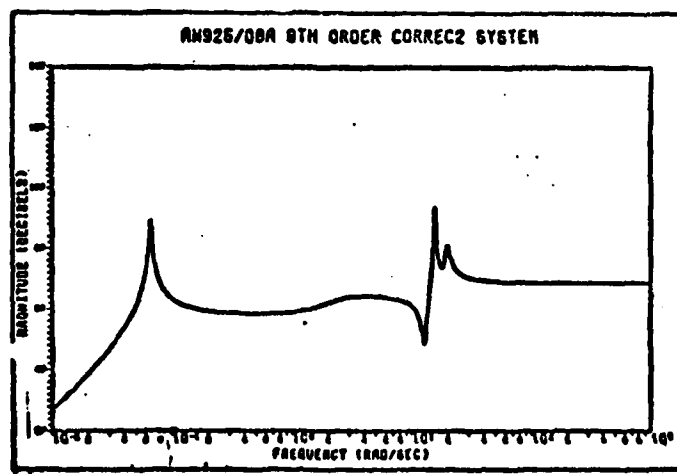




a. Full Order System

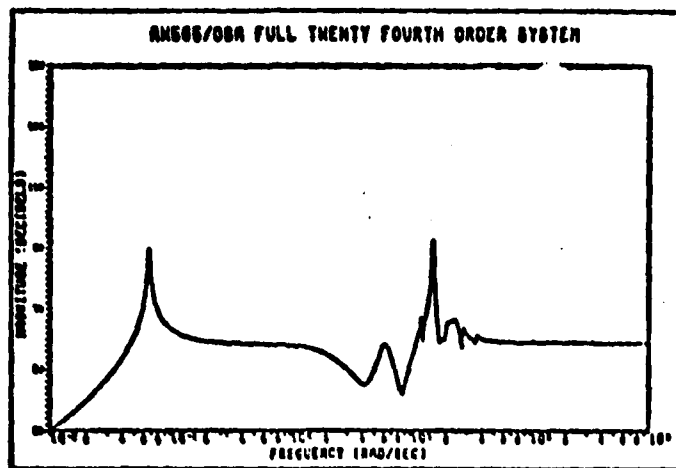


b. 10th Order System (step balanced)

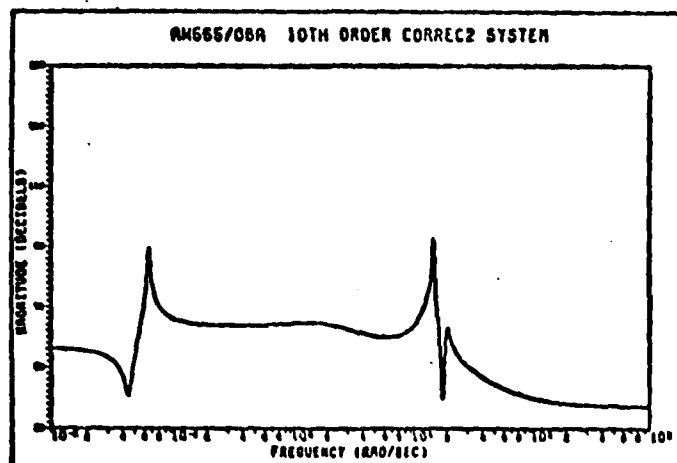


c. 9th Order System (step balancing)

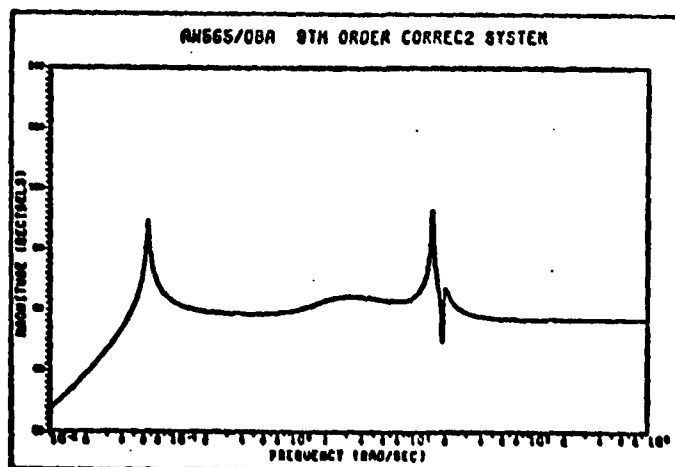
FIGURE 25. Frequency Response (AW925/OBA)
Using Step Balancing



a. Full Order System

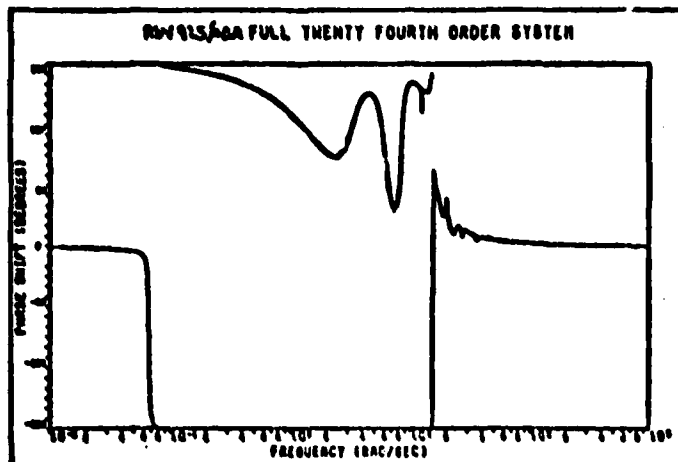


b. 10th Order System (step balanced)

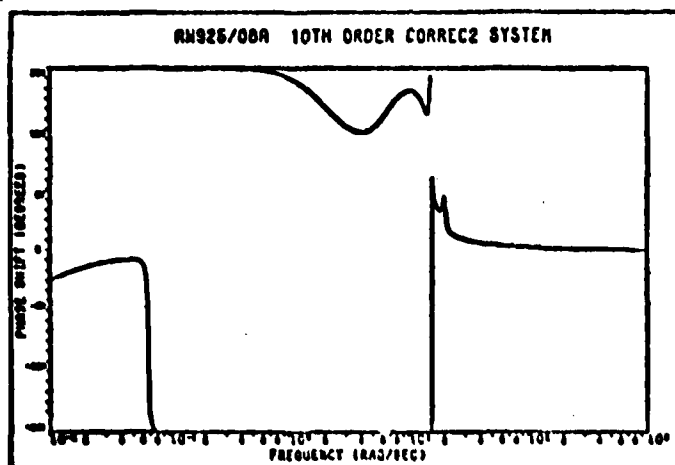


c. 9th Order System (step balanced)

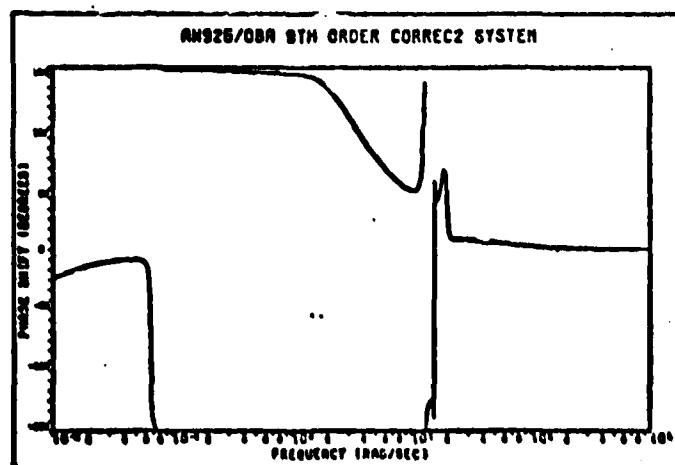
FIGURE 26. Frequency Response (AW565/OBA)
Using Step Balancing



a. Full Order System

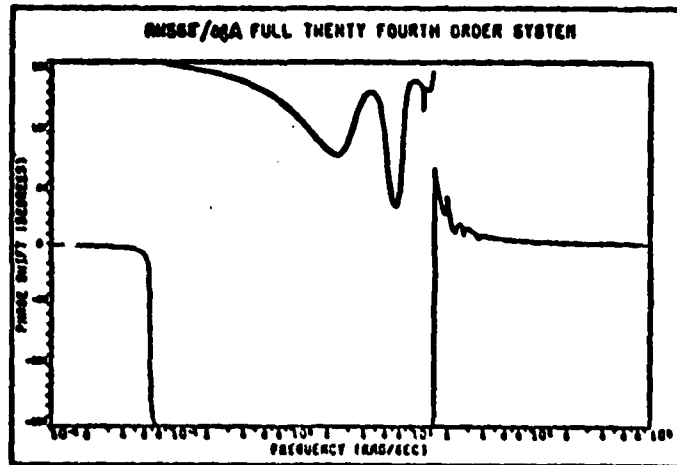


b. 10th Order System (step balanced)

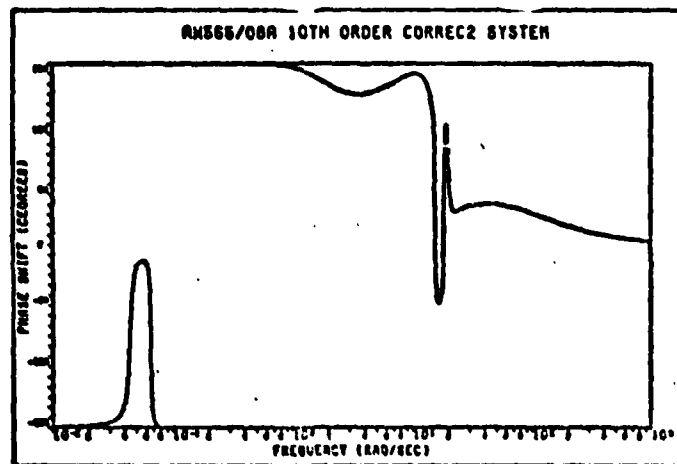


c. 9th Order System (step balanced)

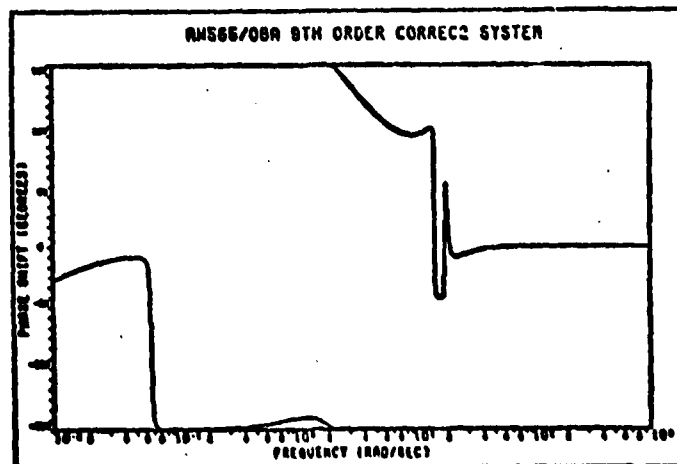
FIGURE 27. Phase Response (AW925/OBA)
Using Step Balancing



a. Full Order System



b. 10th Order System (step balanced)



c. 9th Order System (step balanced)

FIGURE 28. Phase Response (AW565/OBA)
Using Step Balancing

much more representative of the original system than the tenth order system obtained via impulse balancing (Figures 15d, 16d). In addition to improvement in results by using step balancing, the H_{INF} matrix also gives a better guideline for the choice of the lowest order reduced order model than it did for impulse balancing.

H_{INF} Matrix Investigation. Table VII contains the singular values of the H_{INF} matrix for step balancing. Grouping these singular values into "large" and "small" groups yields approximately two singular values in the "large" group. If adhered to strictly, this suggests second order as being the lowest order attainable without severe loss of reproduction accuracy. But if followed as a guideline, possibly fifth or sixth order is the lowest order achievable (subjective).

TABLE VII
Singular Values of the H_{INF} Matrix for Step Balancing

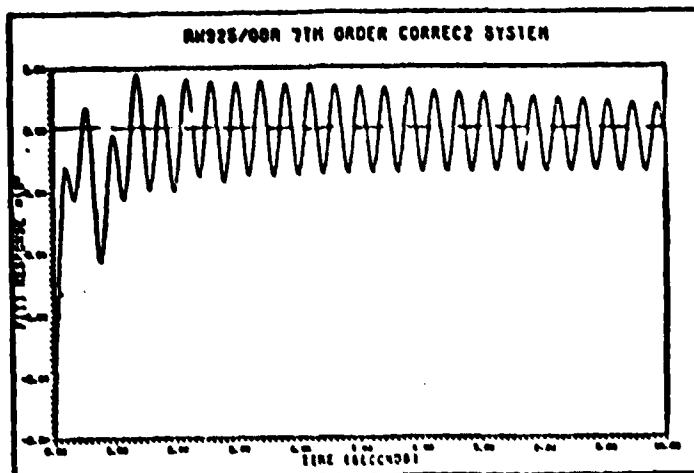
3.3665E+06	6.2795E+02	2.5876E+02	2.7487E+01
3.3625E+06	5.9453E+02	2.3815E+02	2.7123E+01
8.8716E+03	3.3029E+02	1.2885E+02	8.1539E+00
8.8448E+03	3.0580E+02	1.1655E+02	8.0367E+00
6.0047E+03	2.8524E+02	8.1622E+01	2.5982E+00
9.0330E+02	2.7912E+02	8.0676E+01	2.5818E+00

Figure 29 shows the seventh, sixth, and fourth order systems for AW925 in response to a .1 radian step in outboard aileron. Figures 30 and 31 show the frequency and phase responses for AW925/OBA respectively. Highly evident in Figures 30 and 31 is the fact that as the system order is reduced, more and more high frequency information is lost. This is occurring at lower frequencies as the system order is reduced.

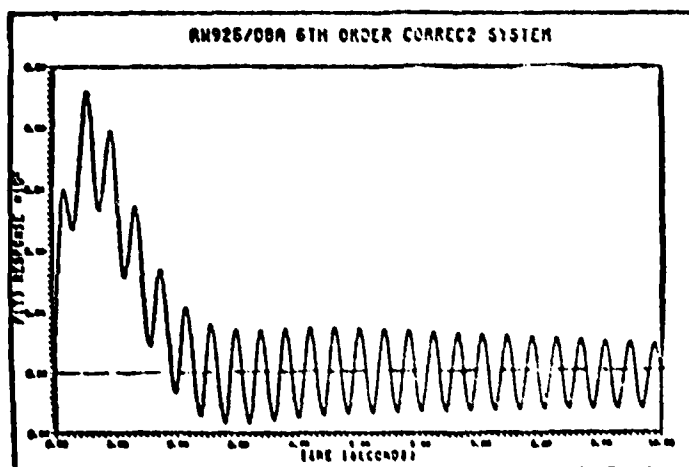
Therefore, depending on an observer's idea of "accurate" reproduction of the original system, many different system orders are achievable. However, approximately tenth order is the lowest order system attainable while still reproducing both the transient and steady state portions of the response fairly well.

Summary

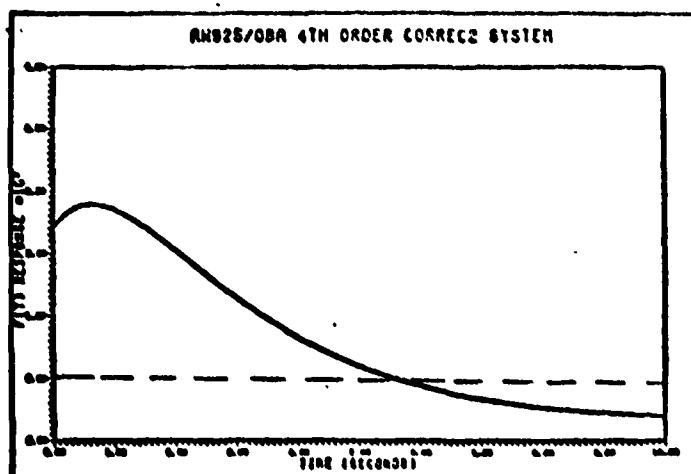
The Moore algorithm seems to provide a viable method for obtaining a reduced order model for a given input system. Rather than delete nondominant poles (if any exist), as many algorithms do, the poles and zeros as discussed in Section II are shifted to compensate for the loss of a state. The Moore algorithm in theory is purging the model of the uncontrollable, unobservable portion of the system as system order is reduced. In practice, the Moore algorithm is discarding high frequency information as system order is reduced.



a. 7th Order System

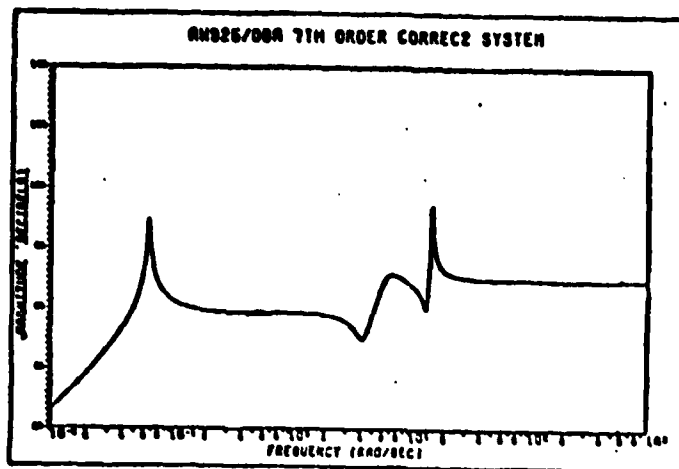


b. 6th Order System

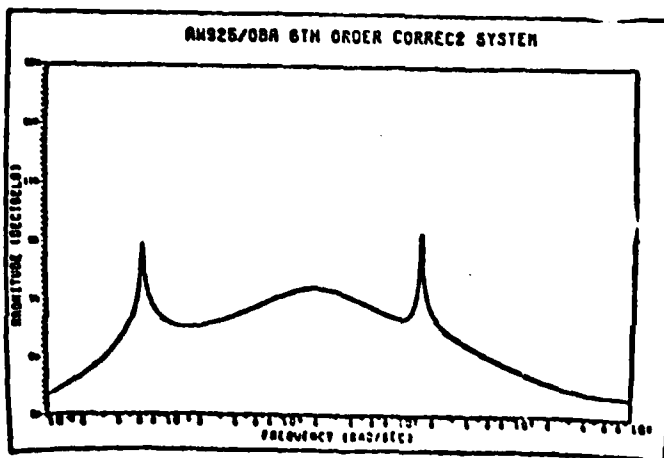


c. 4th Order System

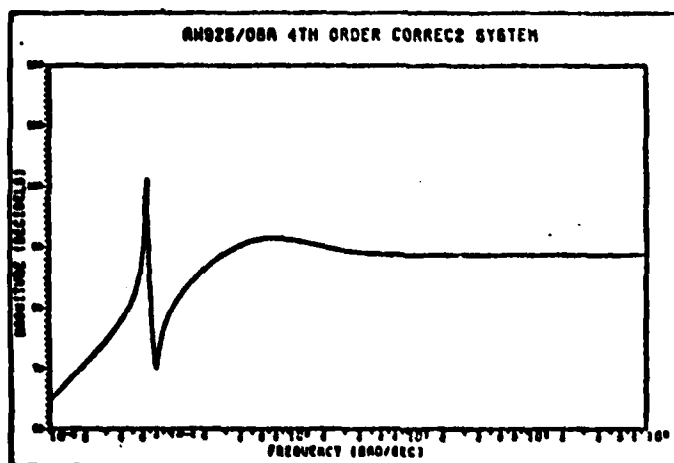
FIGURE 29. AW925/OBA To .1 Rad. Step Input
Using Step Balancing



a. 7th Order System

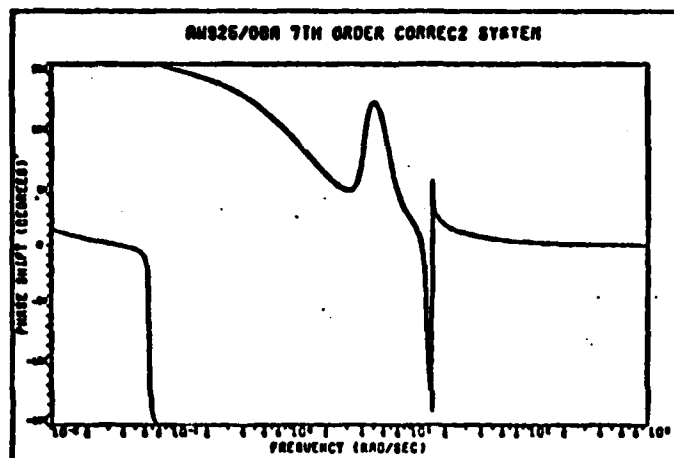


b. 6th Order System

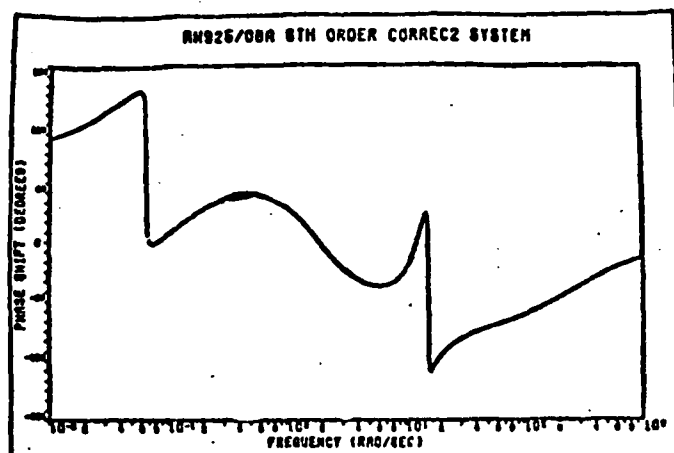


c. 4th Order System

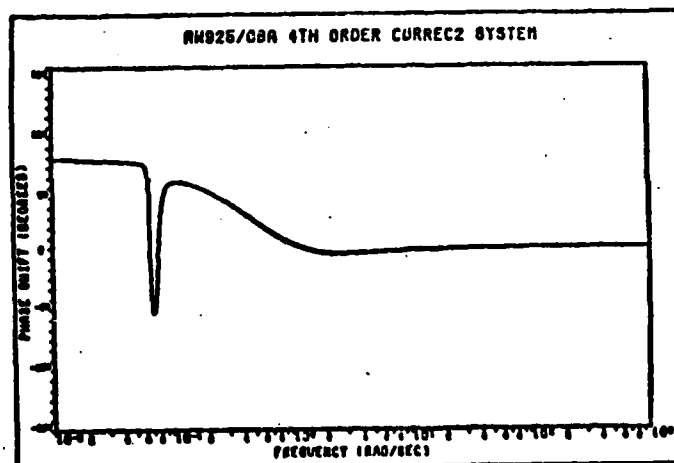
FIGURE 30. Frequency Response (AW925/OBA)
Using Step Balancing



a. 7th Order System



b. 6th Order System



c. 4th Order System

FIGURE 31. Phase Response (AW925/OBA)
Using Step Balancing

The algorithm is easily programmable and the highly efficient subroutine packages for singular value decomposition are readily available and numerically very stable (Ref 13, 36). Potential applications of the Moore algorithm include simulation, on-line control, and Kalman filter design. The derivation of a reduced order model requires no human trial and error process.

The H_{INF} matrix gives a general guideline in the choice of the minimum order for the reduced order model to "accurately" reproduce the original, full-order system.

Although the steady state controllability and observability grammians, which were used exclusively for this study, only allow application of the Moore algorithm to stable, time-invariant systems, implementation of the finite interval grammians would allow application of the Moore algorithm to unstable and time-varying systems.

The Moore algorithm can be used when other classes of inputs are utilized. The lowest achievable order for inputs other than an impulse or step will probably not be as low as if the inputs had been an impulse or step however. This is because the algorithm is "optimized" for impulse and step inputs.

Properties of the Moore algorithm presented in Section II, this section, Appendix B, and References 24 and 25 make the "internally balanced" state coordinate system highly desirable. The internally

balanced state coordinate system is the decomposition of Kalman achieved by using "working subspaces" provided by the controllability and observability grammians. The states are ordered with respect to their controllability and observability properties.

Finally, the Moore algorithm may never be the best model order reduction procedure that can be used, but it does provide a fast and accurate determination of a reduced order model requiring no human trial and error process.

VI. RESULTS AND RECOMMENDATIONS

The goals of this study were twofold. First, the desire was to investigate the Moore algorithm through its application to the relatively high order B-52E CCV flutter control problem. Second, it was desired to develop an interactive computer program which enables a user to not only obtain the balanced representation (via the Moore algorithm) for his system, but provides other options useful in the study of multi-input-multi-output systems as well. This section describes to what extent these objectives were met and the results thereof, as well as recommends areas for future investigation.

SUMMARY OF RESULTS

There are four main results of this study. These results include: successful determination of a tenth order model via the Moore algorithm to compare with AFFDL's tenth order model for the B-52E CCV flutter control problem; successful investigation of the singular values of the H_{∞} matrix as a means of determining the minimum dimension of a reduced order system for "accurate" full order system reproduction; and a working, interactive computer program together with a fully documented user's manual for the program. Each result will now be discussed individually.

Comparison of reduced order models for B-52E CCV flutter control problem. In Section V the Moore algorithm was applied to the B-52E CCV flutter control problem's 24th order model. Figures 23, 24 showed the tenth and ninth order systems obtained via the Moore algorithm vs. the tenth order system obtained via the modified Schwendler and MacNeal technique used by AFFDL. The results are good. The phase shift and damping changes observed in AFFDL's response are not present in the Moore algorithm responses. Since the definition of "goodness" is usually user-defined, it is up to the reader to determine which technique provides the better results. At the very least, however, the Moore algorithm offers desirable properties that the Schwendler and MacNeal technique does not. It is fast, accurate and efficient where AFFDL's technique may be more accurate (dependent on user definition), but is definitely much slower in terms of user development time than the Moore algorithm.

Each technique has its own advantages and disadvantages. AFFDL's technique deals with the "physical" state coordinate system and thus results may be interpreted in a physical way. The Moore algorithm works solely in a mathematical coordinate system and thus no such physical interpretations can be made.

The Moore technique also offers a guideline to the choice of minimum order for a reduced order model. The Moore technique

is applicable to linear, time-invariant, stable, MIMO, or SISO systems. The technique could be applied to unstable systems as well, by either:

1) numerically integrate to solve for the controllability and observability grammians over finite time intervals.

2) obtain an algorithm which solves the two matrix Riccati equations for an A matrix with negative and positive eigenvalues.

These matrix Riccati equations are (Ref 20, 24, 25):

$$A W_{C, \infty}^a + W_{C, \infty}^a A^t = -BB^t \quad (73)$$

$$A^t W_{O, \infty}^a + W_{O, \infty}^a A = -C^t C \quad (74)$$

The Moore algorithm may possibly never be the best of all possible algorithms that exist, but it does provide an accurate reduced order model swiftly and efficiently.

H₂/H_∞ matrix singular values investigation. Section V presented the singular values of the H₂/H_∞ matrix for both impulse balancing and step balancing. The results showed that the H₂/H_∞ matrix singular values do give a fairly accurate guideline to follow in determining the minimum acceptable order of the reduced order system.

MIMO -- an interactive computer program. Section III presented the interactive computer program MIMO. Although time did not allow the development of an extensive error checking capability, the program does function as designed and offers fifteen

options that aid a knowledgeable user in the analysis of multi-input-multi-output systems.

The computer program offers the following options:

- 0) List all available options.
- 1) Terminate MIMO--update local file MEMORE.
- 2) Input system. Obtain impulse balanced or step balanced coordinate system.
- 3) Discretize the input system. Obtain F,G, controllability, observability, and Hankel matrices.
- 4) Plot and optionally list singular values and their ratios vs. sample time for controllability matrix, observability matrix, and/or Hankel matrix.
- 5) Obtain continuous time controllability and observability matrices.
- 6) Estimate the condition number of a square matrix.
- 7) Obtain singular values and/or right and left singular vectors of a matrix.
- 8) Plot and optionally list the frequency response of user-specified order of balanced model.
- 9) Dispose plots to AFTT plotter.
- 10) Update local file MEMORE.
- 11) Recover information from MEMORE.
- 12) Interface to TOTAL.

- 13) Obtain output normal state coordinate system.
- 14) Obtain output predictive state coordinate system.
- 15) Obtain the controllability and observability grammians for the input system.

User's manual. The user's manual is an important part of the development of the computer program. Examples were included to illustrate the program's options. The explanations of each option were made as clear as possible bearing in mind the information to be presented. The assumption was made that the potential user knew how to login to the interactive terminal. Beyond that assumption, the user's manual explains all other steps up until logout. The user's manual is found in Appendix C.

RECOMMENDATIONS FOR FUTURE STUDY

There are several ways that this study can be expanded upon. Each recommendation is now presented.

As mentioned in Section V, the Moore algorithm obtains a system which is robust to perturbations in system elements. Therefore, a useful and interesting follow-on would be to create model mismatch by perturbing the eigenvalues of the system by given percentages. The results should indicate a correlation between the condition number and the amount of model mismatch that can be tolerated for a particular system design.

Another interesting application would be to extend the algorithm to discrete-time systems. The continuous time H_{INF} matrix could possibly be replaced by the finite dimension discrete-time Hankel matrix in the existing algorithm. The two grammians that must be solved would then have to be numerically integrated. Other changes would also probably have to be made.

Many other options could be added to MIMO. With the complexity of systems today, an interactive computer package is a must. New routines could be highly useful in the analysis and design of a given system. Also, an efficient error checking capability coupled with an improved user interface would be highly beneficial.

The Moore algorithm as currently programmed is not as efficient in terms of computer resources required, as possible. The efficient coding of this algorithm by someone well versed in that area for potential on-line applications would be a very desirable follow-on. Analyses could then be done to determine which algorithm out of a field of algorithms had the most potential for on-line applications. These are but a few of the areas that could be investigated.

Since there is a great need for efficient model reduction algorithms, it is hoped that the results of this study will be useful in the determination of such algorithms.

BIBLIOGRAPHY

1. Anderson, L. R. Decoupling and Order Reduction of Two Time-Scale Control Systems. Aerospace and Ocean Engineering Dept. Virginia Polytechnic Institute and State University, Blacksburg, Virginia.
2. Antoh, H. Elementary Linear Algebra. New York: John Wiley & Sons Inc., 1977.
3. Appiah, R.K. "Linear Model Reduction Using Hurwitz Polynomial Approximation", International Journal of Control, Vol. 28, No. 3, 1978.
4. ASD Computer Center. ASD Computer Center Calcomp Plotter Guide (Revision A). Wright-Patterson AFB, Ohio: ASD Computer Center 1976.
5. ASD Computer Center. ASD Computer Center CDC NOS/BE User's Guide (Revision A). Wright-Patterson AFB, Ohio: ASD Computer Center, 1976.
6. Chidambarra, M. R. "On a Method for Simplifying Large Linear Dynamic Systems", IEEE Tran. on Auto. Control. Vol. AC-12, February 1967.
7. Cline, A. K. et al. "An Estimate for the Condition Number of a Matrix", Siam Journal of Numerical Analysis. Vol. 16, No. 2, 1979.
8. Control Data Corporation. Fortran Extended Version 4 Reference Manual (Revision B). Sunnyvale, California: Control Data Corporation, 1976.
9. Control Data Corporation. Intercom Version 4 Reference Manual (Revision B). Sunnyvale, California: Control Data Corporation, 1976.
10. Control Data Corporation. Loader Version 1 Reference Manual (Revision J). Sunnyvale, California: Control Data Corporation, 1976.
11. Dazzo, J. J., and C. H. Houps. Linear Control System Analysis and Design: Conventional and Modern. New York: McGraw-Hill Book Co., 1975.

12. Derusso, P. M., R. J. Roy, and C. M. Close. State Variables for Engineers. New York: John Wiley & Sons Inc., 1965.
13. Dongarra, J. J. et al. LINPACK User's Guide. Philadelphia: Siam, 1979.
14. El-Attar, R. A., and M. Vidyasagar. "Order Reduction by l_1 and l_∞ -Norm Minimization", IEEE Trans. Auto. Control, Vol AC-23, No. 4, 1978.
15. Forsythe, G., Malcom, M. A., and C. B. Moler. Computer Methods for Mathematical Computations. Prentice-Hall, Englewood Cliffs, New Jersey, 1977.
16. Gantmacher, F. R. The Theory of Matrices. Chelsea Publishing Co., New York, 1964.
17. Hutton, M. F., and B. Friedland. "Routh Approximations for Reducing Order of Linear, Time-Invariant Systems", IEEE Trans. Auto. Control, Vol. AC-20, No. 3, 1975.
18. Hutton, M. F., and M. J. Rabins. "Simplification of High-Order Mechanical Systems Using the Routh Approximation", Journal of Dynamic Systems, Measurements, and Control, Vol. 97, 1975.
19. Kalman, R. E. "Mathematical Description of Linear Dynamical Systems", J. SIAM Control, Vol. 1, No. 2, 1963.
20. Kleinman, D. L. A Description of Computer Programs for Use in Linear Systems Studies. TR-77-2. Storrs, Connecticut: School of Engineering, University of Connecticut, 1977.
21. Larimer, S. J. An Interactive Computer Aided Design Program for Digital and Continuous Control System Analysis and Synthesis. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, March 1978. (AD A055 418)
22. Maybeck, P. S. Professor, Air Force Institute of Technology, Wright-Patterson AFB, Ohio. (Personal interview), October 31, 1979.
23. Melsa, J. L., and S. K. Jones. Computer Programs for Computational Assistance in the Study of Linear Control Theory (Second Edition). New York: McGraw Hill Book Co., 1973.

24. Moore, B.C. Singular Value Analysis of Linear Systems, Part I: External Variables. Systems Control Report No. 7801, Dept. of Electrical Engineering, University of Toronto, Toronto, Ontario. (1978)
25. Moore, B.C. Singular Value Analysis of Linear Systems, Part II: Controllability, Observability, and Model Reduction. Systems Control Report No. 7802, Dept. of Electrical Engineering, University of Toronto, Toronto, Ontario. (1978)
26. O'Brien, F.L. A Consolidated Computer Program for Control System Design. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1976. (AD A034 879)
27. Reddy, A.S.S.R. "A Method for Frequency Domain Simplification of Transfer Functions", International Journal of Control, Vol. 23, 1976.
28. Reid, J.G. Linear Systems Analysis and Digital Computation Methods. (Unpublished Course Notes) Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology.
29. Reid, J.G. Assistant Professor, Air Force Institute of Technology, Wright-Patterson AFB, Ohio. (Personal interview) April 17, 1979.
30. Reid, J.G., et al. "Robustness Properties of Output Predictive Dead-Beat Control: SISO Case". To be published. Air Force Institute of Technology, Wright-Patterson AFB, Ohio. (1979)
31. Schwanz, R.C. Analytical Flight Test Measurements for Estimation of Aerodynamic Parameters of Flexible Aircraft. AFITDL TM-78-104-FGC. Wright-Patterson AFB, Ohio: AF Flight Dynamics Laboratory, 1978.
32. Schwendler, R.G., and R.H. MacNeal. Optimum Structural Representation in Aeroelastic Analyses. ASD-TR-61-680. Wright-Patterson AFB, Ohio: Flight Control Laboratory, 1962.
33. Shamash, T. "Model Order Reduction Using the Routh Stability Criterion and the Pade' Approximation Technique", International Journal of Control. Vol. 21, No. 3, 1975.

34. Shieh, L. S., and M. J. Goldman. "A Mixed Cauer Form for Linear System Reduction", IEEE Tran. On Systems, Man, and Cybernetics. November 1974.
35. Skelton, R. E. "Observability Measures and Performance Sensitivity in the Model Reduction Problem", International Journal of Control. Vol. 29, No. 4, 1979.
36. Stewart, G. W. Introduction to Matrix Computations. New York: Academic Press, 1973.

APPENDIX A

DEFINITION OF THE SINGULAR VALUE DECOMPOSITION

APPENDIX A

DEFINITION OF THE SINGULAR VALUE DECOMPOSITION

A singular value decomposition occurs when a general matrix is reduced to diagonal form by premultiplying and postmultiplying it by unitary matrices (Ref 36: 318-325, Ref 13: 11.1-11.3, Ref 24: 6-8).

The equation is written as:

$$V^H A_{m \times p} U = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} ; \text{ where } V^H \text{ denotes the conjugate transpose of } V \quad (A-1)$$

where A is a general matrix

$$\Sigma_1 = \text{diag} (\sigma_1, \sigma_2, \dots, \sigma_N) \text{ and } \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$$

$$n > r$$

$$r = \min (m, p)$$

V, U are unitary matrices

The columns of V are called the left singular vectors. They are the eigenvectors of AA^H . The columns of U are called the right singular vectors. They are the eigenvectors of $A^H A$. The singular values; $\sigma_1, \sigma_2, \dots, \sigma_r$, are the non-negative square roots of the eigenvalues of $A^H A$.

Thus, any general matrix can be factored into:

$$A_{m \times p} = V_{m \times m} \Sigma_{m \times p} U_{p \times p}^H \quad (A-2)$$

This factorization of a matrix has several "nice" properties. Since only unitary matrices are used, errors are not magnified. Thus, the $r \times r$ sub-matrix Σ_1 will only be of defective rank if the matrix is of defective rank (Ref 36:318).

The left singular vectors are the axes of an ellipsoid with the corresponding singular values being the lengths of the axis of the ellipsoid. Thus $\sigma_1 = \sigma_{\max}$ is the length of the major axis, and $\sigma_r = \sigma_{\min}$ is the length of the minor axis of the ellipsoid.

Defining a quantity, $K(A)$, as the ratio of $\sigma_{\max}$ to $\sigma_{\min}$ ($\sigma_{\max}/\sigma_{\min}$), we have a measure of eccentricity of the ellipsoid. The scalar $K(A)$ is termed the "condition number with respect to inversion" of the matrix A . The larger in magnitude the condition number, the more eccentric (near degenerate) is the ellipsoid, and thus, the more closely the A matrix is to being singular. Therefore, unlike eigenvalues which tell us only if a matrix is singular or not, singular values give us an idea of how close we are to being singular (i.e., how big the perturbations in the elements of the matrix can be before the matrix becomes singular).

$$\text{Defining } \|A\|_2 = \max_{\|x\|_2 = \|y\|_2 = 1} |y^T A x|$$

(A-3)

as the 2- norm of a matrix (Ref 36:180), $\sigma_1 = \sigma_{\max}$ is indeed the value of the norm; i.e., $\|A\| = \sigma_1$. Likewise, $\|A^+\| = 1/\sigma_r$; where A^+ denotes the generalized inverse. (A-4)

Finally, $K(A) = \sigma_{\max}/\sigma_{\min}$ also equals $\|A\| \|A^+\|$, which is the formal definition of $K(A)$ (Ref 7:1).

Singular values have many other properties which are important, but are beyond the scope of this appendix. The reader is referred to Reference 36:318-325. The material presented in this appendix hopefully will acquaint the reader who has a linear algebra background with enough of the basics of the singular value decomposition to follow the development in the text.

APPENDIX B

DESCRIPTION OF THE ALGORITHM

(Ref 24, 25)

APPENDIX B

DESCRIPTION OF THE ALGORITHM (Ref 24, 25)

The algorithm to be presented here was developed by Dr Bruce Moore of the University of Toronto. The algorithm will be developed and presented to enhance the understanding of the properties and results from the application of this algorithm. A more extensive development may be found in References 24 and 25.

To comprehend the development of this algorithm, one requires an understanding of singular value analysis. A brief description of singular value analysis is presented in Appendix A. The reader is also referred to Reference 36 for an excellent treatment of the subject.

Singular value analysis is applied to linear static equations, followed by its application to linear differential equations. The results will then be applied to state space models yielding the "internally balanced" representation, which will result in reduced order models. Results for the algorithms as applied to a third order example as described by El-Attar and Vidyasagar in Reference 14 are presented in Section IV of this thesis.

The following discussion is divided into two major areas (Ref 24, 25):

- 1) external variable analysis (those which may be measured and/or manipulated),

2) internal variable analysis (state variable coordinate system).

Both sets of variables need to be considered for an effective model reduction technique.

The notation will be the same as Dr. Moore used in References 24 and 25 to enable the reader to easily make the transition between this paper and Dr. Moore's papers. The notation is:

R	The field of real numbers
$C[0, t_1]$	ring of piecewise continuous functions on interval $[0, t_1]$
R^m	m^{th} dimensional vector space
$C^m[0, t_1]$	m^{th} dimensional vector space
$\text{Ker}(M)$	Null space of M
$\text{Im}(M)$	Range space of M
M^T	transpose of M
S	Subspace
$S^\perp, \delta R^N$	orthogonal complement in R^N
Capital letters	Matrices and maps
Underscored letters	Vectors
$\ \underline{v}\ $	Euclidean norm of a vector
$\ M\ $	Subordinate Spectral norm for a matrix

$$\text{where } \|M\| = \max_{\|\underline{x}\| = 1} \|M\underline{x}\|$$

For the linear equation

$$\underline{n}_r = M_{rxm} \underline{w}_m \quad (B-1)$$

where M can be thought of as a map transforming R^m into R^r (Ref 28:77-78), there are four subspaces which completely characterize equation (B-1). These subspaces are:

$$\text{Im}(M)$$

$$\text{Im}(M)^\perp = \text{Ker}(M^T)$$

$$\text{Ker}(M)$$

$$\text{Ker}(M)^\perp = \text{Im}(M^T).$$

For a non-trivial solution ($\underline{w} \neq \underline{0}$) to exist, $\text{Ker}(M) \neq \underline{0}$. The columns of M must be linearly dependent (Ref 28:3-71). Then by defining V_1 and V_2 to be matrices with orthogonal columns (basis vectors) which span $\text{Ker}(M)$ and $\text{Ker}(M)^\perp$, respectively, we may write the vector $\underline{w}$ as

$$\underline{w} = V_1 \underline{\hat{w}}_1 + V_2 \underline{\hat{w}}_2. \quad (B-2)$$

Substituting equation (B-2) into equation (B-1) and rewriting yields

$$\underline{n} = M V_1 \underline{\hat{w}}_1 + M V_2 \underline{\hat{w}}_2 \quad (B-3)$$

Since $\text{Ker}(M)$ is comprised of the subspace of all possible solutions to the homogeneous linear equation

$$M \underline{w} = \underline{0} \quad (\text{Ref 28:3-58}), \quad (B-4)$$

$MV_2 = \underline{0}$. Therefore, equation (B-2) reduces to

$$\underline{n} = MV_1 \underline{w}_1 \quad (B-5)$$

Thus, it is seen that the variable $\hat{\underline{w}}_2 = V_2^T \underline{w}$ has no effect on the system. A redundancy exists and is stripped away by reduction of equation (B-1) to equation (B-5).

Similarly, if we define U_1, U_2 as two matrices whose orthonormal columns span $\text{Im}(M)$ and $\text{Im}(M)^\perp$, respectively, then equation (B-1) can be rewritten as

$$U_1^T \underline{n} + U_2^T \underline{n} = U_1^T M \underline{w} + U_2^T M \underline{w}, \quad (B-6)$$

which reduces to

$$U_1^T \underline{n} = U_1^T M \underline{w}; \text{ where } U_2^T \underline{n} = \underline{0} \quad (B-7)$$

This exposes the redundancy associated with $\underline{n}$.

Combining the results of equations (B-5) and (B-7), thereby removing the redundancy yields

$$U_1^T \underline{n} = (U_1^T M V_1) \hat{\underline{w}}_1. \quad (B-8)$$

The minimum norm vector $\underline{w}$ which minimizes $\|\underline{R} - M \underline{w}\|$ is

$$\underline{w} = V_1 \hat{\underline{w}}, \quad (B-9)$$

where $\hat{\underline{w}}$ satisfies

$$U_1^T \underline{n} = (U_1^T M V_1) \hat{\underline{w}} \quad (B-10)$$

Unfortunately, though theoretically sound, the above computations might yield erroneous results when programmed into a digital computer.

Fixed wordlengths, truncation and the like can cause a matrix of less than full rank to appear to be of full rank. However, through use of the singular value decomposition these restrictions' effects are greatly reduced.

If we define the following subspaces

$$\begin{aligned} S_m &= \{ \underline{n} \in R^r : \underline{n} = M \underline{w}, \|\underline{w}\| = 1 \} \\ S_m^T &= \{ \underline{w} \in R^m : \underline{w} = M^T \underline{n}, \|\underline{n}\| = 1 \} \end{aligned} \quad (B-11)$$

which are contained in $\text{Im}(M)$, $\text{Ker}(M)$, respectively, and perturb M by no more than $\|\Delta M\|$ ($M_\Delta = M + \Delta M$), then $S_{m\Delta}$, $S_{m\Delta}^T$ must be "nearly" in those subspaces.

Referring to Appendix A, the singular value factorization of a matrix M yields

$$M = U_1 \Sigma_1 V_1^T. \quad (B-12)$$

U_1 and V_1 have orthonormal columns which span $\text{Im}(M)$ and $\text{Ker}(M)^\perp$ respectively.

Then, by defining U_2 , V_2 to be matrices with orthonormal columns which span $\text{Im}(M)^\perp$, $\text{Ker}(M)$, respectively, the singular value factorization of M may be rewritten as

$$M = (U_1 U_2) \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U \Sigma V^T. \quad (\text{B-13})$$

The columns of U are the left singular vectors, while the columns of V are the right singular vectors. We can rewrite $\underline{n} = M\underline{w}$ now as

$$\begin{bmatrix} \underline{\bar{n}}_1 \\ \underline{\bar{n}}_2 \end{bmatrix} = \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} M\underline{w} = \begin{bmatrix} \Sigma_1 & V_1^T & \underline{w} \\ 0 & & \end{bmatrix} \quad (\text{B-14})$$

where $\underline{\bar{n}}_1 = U_1^T \underline{n}_1$
 $\underline{\bar{n}}_2 = U_2^T \underline{n}_2.$

Thus we can obtain

$$\Sigma_1^{-1} \underline{\bar{n}}_1 = V_1^T \underline{w}$$

$$\|\Sigma_1^{-1} \underline{\bar{n}}_1\|^2 = \|V_1^T \underline{w}\|^2 \leq \|\underline{w}\|^2 \quad (\text{B-15})$$

Equation (15) allows us to view the singular value factorization from a geometrical point-of-view. For, every vector in S_M is contained an elliptical region in R^T . The boundary of which is defined by (Ref 24:7)

$$\frac{\bar{n}_{1,1}^2}{\sigma_{m,1}^2} + \dots + \frac{\bar{n}_{1,l}^2}{\sigma_{m,l}^2} = 1. \quad (\text{B-16})$$

where

$$\sigma_{m,i} \geq \sigma_{m,i+1}$$

As we saw in equations (B-6) and (B-7), U_2 is associated with redundancy as applied to $\underline{n} = M\underline{w}$. Therefore, the column vectors of U_2 are unit vectors aligned with a degenerate axis of an ellipse, where its corresponding singular value is the length of that axis. Just the opposite, the columns of U_1 are the nondegenerate axes of the ellipses.

Hence, the degenerate axis of each ellipse even under slight perturbation spans a space very close to $\text{Im}(M)^\perp$. The nondegenerate axis of each ellipse spans a space very close to the $\text{Im}(M)$.

Similarly, the subspace S_{in}^T can be so characterized. Let the columns of V be the coordinate vectors $\in R^m$, so that $\underline{n} = M\underline{w}$ becomes

$$\underline{n} = M(V_1 V_2) \begin{bmatrix} \hat{w}_1 \\ \hat{w}_2 \end{bmatrix} = U_1 \Sigma_1 \hat{w}_1. \quad (\text{B-17})$$

We can write

$$\|\Sigma_1 \hat{w}_1\|^2 = \|U_1^T \underline{n}\|^2 = 1 \quad (\text{B-18})$$

Thus the equation of the ellipse can now be written as (Ref 24:8)

$$\frac{\hat{w}_{1,1}^2}{(1/\sigma_{m,1})^2} + \dots + \frac{\hat{w}_{1,l}^2}{(1/\sigma_{m,l})^2} = 1 \quad (\text{B-19})$$

From Appendix A we know that

$$\sigma_1 = \sigma_{\max} = \|M\|. \quad (\text{B-20})$$

Using the equation $M_\Delta = M + \Delta M$, we can define an upperbound on the effect of a perturbation on M by writing

$$|\sigma_{m,1} - \sigma_{m_\Delta,1}| = \|\Delta M\| = \sigma_{\Delta m, \max}. \quad (\text{B-21})$$

Then generalizing, equation (B-21) can be rewritten as

$$|\sigma_{m,i} - \sigma_{m_\Delta,i}| \leq \|\Delta M\|. \quad (\text{B-22})$$

Singular values are also useful in the determination of the worst case sensitivity of minimum norm solutions of $\underline{n} = M\underline{w}$ to perturbations in $\underline{n}$.

Defining $\underline{w}$, $\underline{w}_\Delta$ to be minimum norm vectors satisfying $\underline{n} = M\underline{w}$, such that the matrices U and V in the singular value factorization of M are the Identity matrix, the following equation can be written.

$$\frac{\|\Delta \underline{w}\|}{\|\underline{w}\|} = \frac{\sigma_{m,1}}{\sigma_{m,l}} \frac{\|\Delta \underline{n}\|}{\|\underline{n}\|}. \quad (\text{B-23})$$

where if M is $n \times n$, $\sigma_{m,l}/\sigma_{m,s}$ is the condition number with respect to inversion (Ref 24:9).

The previous results show us that though the singular value factorization of M is

$$M = U_1 \Sigma_1 V_1^T = [U_{1,L} \ U_{1,s}] \begin{bmatrix} \Sigma_{1,L} & 0 \\ 0 & \Sigma_{1,s} \end{bmatrix} \begin{bmatrix} V_{1,L}^T \\ V_{1,s}^T \end{bmatrix}, \quad (B-24)$$

where L denotes "large", and s denotes "small", can be rewritten as

$$M = U_1 \Sigma_1 V_1^T = U_{1,L} \Sigma_{1,L} V_{1,L}^T \quad (B-25)$$

if the elements in $U_{1,s} \Sigma_{1,s} V_{1,s}^T$ are smaller in magnitude than the uncertainty of the given system.

Now, expanding the previous results to systems of linear differential equations, many of the properties apply. The equation of concern is

$$\underline{\dot{n}}(t) = M(t) \underline{w}(t), \quad 0 \leq t \leq t_1. \quad (B-26)$$

The map $H_{m,t_1} : R^m \rightarrow C^r(0, t_1)$ is defined by equation (B-26).

The map $L_{m,t_1} : C^m(0, t_1) \rightarrow R^r$ is defined by

$$\underline{n}(t_1) = \int_0^{t_1} M(t-\tau) \underline{U}(\tau) d\tau. \quad (B-27)$$

Again, associated with the map $M(t)$ the following fundamental subspaces exist:

$$\text{Im}(L_{m,t_1}) = \{ \underline{n} : \underline{n} = \int_0^{t_1} M(t_1 - \tau) \underline{U}(\tau) d\tau, \underline{U} \in C^m(0, t_1) \} \quad (\text{B-28})$$

$$\text{Ker}(H_{m,t_1}) = \{ \underline{w} : M(t) \underline{w} = 0, 0 \leq t \leq t_1 \}$$

$$\text{Im}(L_{m,t_1})^\perp = \text{Ker}(H_{m,T,t_1})$$

$$\text{Ker}(H_{m,t_1})^\perp = \text{Im}(L_{m,T,t_1})$$

As before, if $\text{Im}(L_{m,t_1})$ does not span R^r (such that a non-trivial solution does exist), then there exists two matrices U_1, U_2 with orthonormal columns which span $\text{Im}(L_{m,t_1}), \text{Im}(L_{m,t_1})^\perp$ so that

$$\int_0^{t_1} U_2^T M(t_1 - \tau) \underline{U}(\tau) d\tau = \underline{0} \text{ for all } \underline{U} \in C^m(0, t_1).$$

Rewriting equations (B-26) and (B-27) yields

$$\begin{aligned} U_1^T \underline{n}(t) &= U_1^T M(t) \underline{w} & 0 \leq t \leq t_1 \\ U_2^T \underline{n}(t) &= \underline{0} \end{aligned} \quad (\text{B-29})$$

and

$$\begin{aligned} U_1^T \underline{n}(t_1) &= \int_0^{t_1} U_1^T M(t_1 - \tau) \underline{U}(\tau) d\tau \\ U_2^T \underline{n}(t_1) &= \underline{0}. \end{aligned} \quad (\text{B-30})$$

Similarly, if $\text{Ker } (m, t_1) = 0$, then two matrices V_1, V_2 exist which span $\text{Ker } (Hm, t_1)^\perp$ and $\text{Ker}(Hm, t_1)$, respectively. Writing

$$\underline{w} = V_1 \hat{\underline{w}}_1 + V_2 \hat{\underline{w}}_2 \quad (\text{B-31})$$

$$\underline{U}(t) = V_1 \hat{\underline{U}}_1(t) + V_2 \hat{\underline{U}}_2(t),$$

where $M(t)V_2 = 0$ for $t \in [0, t_1]$, equations (B-26) and (B-27) can be rewritten as

$$\underline{n}(t) = M(t) V_1 \hat{\underline{w}}_1, \quad 0 \leq t \leq t_1 \quad (\text{B-32})$$

$$\underline{n}(t_1) = \int_0^{t_1} M(t_1 - \tau) V_1 \hat{\underline{U}}_1(\tau) d\tau.$$

Combining equations (B-29) and (B-32) yields

$$U_1^T \underline{n}(t) = (U_1^T M(t) V_1) \hat{\underline{w}}_1, \quad 0 \leq t \leq t_1 \quad (\text{B-33})$$

$$U_1^T \underline{n}(t_1) = \int_0^{t_1} (U_1^T M(t_1 - \tau) V_1) \hat{\underline{U}}_1(\tau) d\tau$$

Therefore, in order to use the results of the previous section, and to be able to program the algorithm on a computer, we need matrix representations of the four subspaces.

Forming the equation

$$W_{m, t_1}^o = \int_0^{t_1} M(t) M^T(t) dt \quad (\text{Grammian}), \quad (\text{B-34})$$

that important matrix representation is achieved. This matrix is W_{m, t_1} , which is the symmetric square root of equation (B-34).

Then $\text{Im}(L_m, t_1) = \text{Im}(W_m, t_1)$, and $\text{Ker}(L_m, t_1) = \text{Ker}(W_m^T, t_1)$. Application of singular value analysis to $W_m, t_1(W_m^T, t_1)$ will reflect the norm characteristics of the maps with no distortion.

This allows the geometric concept obtained previously to be extended to linear differential equations. The reader should note that W_m, t_1 is symmetric and positive definite. Similar to equation (B-22), equation (B-35) can be written as

$$|\sigma_{1,i}^2 - \sigma_{2,i}^2| \leq \left\| \int_0^{t_1} \Delta M^T M(t) dt \right\|. \quad (\text{B-35})$$

To define the worst case magnification of perturbation in the solution of $\underline{n}(t) = M(t)\underline{w}(t)$, the condition numbers of W_m, t_1 and W_m^T, t_1 may be used.

Moore shows in Reference 24 that if $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$ (singular values of W_m, t_1), $\underline{n} \in \mathbb{R}^r$, then $\underline{U}_n \in C^m(0, t_1)$ be the minimum norm function satisfying $\underline{n} = L_m, t_1(\underline{U}_n)$, produces

$$\left[\frac{\int_0^{t_1} \|\Delta \underline{U}(t)\|^2 dt}{\int_0^{t_1} \|\underline{U}_n(t)\|^2 dt} \right]^{\frac{1}{2}} = \left[\frac{\sigma_1}{\sigma_r} \right] \frac{\|\Delta \underline{n}\|}{\|\underline{n}\|}, \quad (\text{B-36})$$

where

$$\Delta \underline{U}(t) = \underline{U}_{n+\Delta n}(t) - \underline{U}_n(t).$$

Similarly, Moore shows that given $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_m \geq 0$ (singular values of Wm^T, t_1), there exist $\underline{w}, \Delta \underline{w}$ such that

$$\left[\frac{\int_0^{t_1} \|M(t) \Delta \underline{w}\|^2 dt}{\int_0^{t_1} \|M(t) \underline{w}\|^2 dt} \right]^{\frac{1}{2}} = \left[\frac{\sigma_m}{\sigma_1} \right] \frac{\|\Delta \underline{w}\|}{\|\underline{w}\|} \quad (B-37)$$

Extending the previous results of linear static equations once again allows the equations

$$W_{m\Delta, t_1} = (\hat{U}_1 \hat{U}_2) \begin{bmatrix} \hat{\Sigma}_L & 0 \\ 0 & \hat{\Sigma}_S \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} \quad (B-38)$$

$$W_{mT\Delta, t_1} = (\tilde{V}_1 \tilde{V}_2) \begin{bmatrix} \tilde{\Sigma}_L & 0 \\ 0 & \tilde{\Sigma}_S \end{bmatrix} \begin{bmatrix} \tilde{V}_1^T \\ \tilde{V}_2^T \end{bmatrix}$$

to be written. If $\tilde{\Sigma}_S, \hat{\Sigma}_S$ elements are small, $\hat{U}_1, \hat{U}_2, \tilde{V}_1, \tilde{V}_2$ may be used as working subspaces for $\text{Im}(Lm, t_1), \text{Im}(Lm, t_1)^\perp, \text{Ker}(Hm, t_1)^\perp$, and $\text{Ker}(Hm, t_1)$, respectively.

The previous results will now be applied to external variables (those which may be manipulated and/or measured). Here, we can define the system to be $\underline{Y}(t) = M(t)\underline{U}(t)$.

Defining $Wm, t_1 = U_{\text{out}} \Sigma_{\text{out}} U_{\text{out}}^T$ and $Wm^T, t_1 = V_{\text{in}} \Sigma_{\text{in}} V_{\text{in}}^T$ (where "out" and "in" suggest association with outputs and inputs), we may apply orthogonal transformations to yield

$$\hat{\underline{Y}}(t) = U_{\text{out}}^T \underline{Y}(t), \quad \underline{U}(t) = V_{\text{in}} \hat{\underline{U}}(t). \quad (\text{B-39})$$

Our system is now $\hat{\underline{Y}}(t) = M(t) \hat{\underline{U}}(t)$. Then, $\hat{\underline{Y}}(t) = \hat{M}(t) \hat{\underline{U}}(t) = U_{\text{out}}^T M(t) \hat{\underline{U}}(t) = U_{\text{out}}^T M(t) V_{\text{in}} \hat{\underline{U}}(t)$. (B-40)

Therefore, $\hat{M}(t) = U_{\text{out}}^T M(t) V_{\text{in}}$. Proceeding to obtain the grammians yields

$$W_{\hat{M}, t_1} = \int_0^{t_1} \hat{M}(t) \hat{M}^T(t) dt = \Sigma_{\text{out}}^2 \quad (\text{B-41})$$

$$W_{\hat{M}^T, t_1} = \int_0^{t_1} \hat{M}^T(t) \hat{M}(t) dt = \Sigma_{\text{in}}^2$$

With this result, we may say that the outputs are ordered with respect to responsiveness to an input, and the inputs are ordered with respect to their influence on the output. Unfortunately, however, $W_{\hat{M}, t_1}$, $W_{\hat{M}^T, t_1}$ are dependent on the internal coordinate system.

Therefore, the discussion will now turn to the internal coordinate system. The discussion will use the state space representation. Given equations of the form

$$\dot{\underline{X}}(t) = A \underline{X}(t) + B \underline{u}(t) \quad (\text{B-42})$$

where $\underline{Y}(t) = C \underline{X}(t)$,

$$\underline{X}(t) \in \mathbb{R}^N, \quad \underline{Y}(t) \in \mathbb{R}^m.$$

Let,

$$M_1(t) = e^{At}B, \quad M_2(t) = Ce^{At} \quad (B-43)$$

Then, $\text{Im}(Lm_1, t_1)$ is the controllable subspace X_c . The unobservable subspace is equal to $\text{Ker}(Hm_2, t_1)$. Then, by previous results we can show that $\text{Im}(Lm_1, t_1) = \text{Im}(W_c, t_1)$, and $\text{Ker}(Hm_2, t_1) = \text{Ker}(W_o, t_1)$. W_c, t_1 is the symmetric square root of $W_c^2, t_1 \equiv \int_0^{t_1} e^{At}BB^Te^{A^Tt}dt$. W_o, t_1 is the symmetric square root of $W_o^2, t_1 \equiv \int_0^{t_1} e^{A^Tt}C^TCe^{At}dt$.

Redundancy of state variables is caused by uncontrollable, unobservable modes (Ref 24:25). By applying our geometrical concept provided by singular value analysis to W_c, t_1 and W_o, t_1 , we have an "image" of the controllable (observable) subspace. We would like to then interpret near degeneracy of the ellipse (condition number large) as a measure of redundancy of state variables. Unfortunately, as in equation (41) the controllability and observability grammians are state coordinate dependent.

If we apply a coordinate transformation to (A, B, C) to yield (A', B', C') , we have the following equations:

$$\begin{aligned} A' &= T^{-1}AT \\ B' &= T^{-1}B \\ C' &= CT. \end{aligned} \quad (B-44)$$

The controllability and observability grammians are related by

$$\hat{W}_{c,t_1}^a = T^{-1} W_{c,t_1}^a T^{-1T} \quad (B-45)$$

$$\hat{W}_{o,t_1}^a = T^T W_{o,t_1}^a T.$$

Then performing a singular value factorization on W_{c,t_1} ,

W_{o,t_1} yields

$$W_{c,t_1} = U_{c,t_1} \Sigma_{c,t_1} U_{c,t_1}^T \quad (B-46)$$

$$W_{o,t_1} = U_{o,t_1} \Sigma_{o,t_1} U_{o,t_1}^T$$

The $\cdot, t_1$ notation will now be deleted for simplicity. By letting T^{-1} in equation (B-45) equal $\Sigma_c^{-1} U_c^T$, we get $W_c^a = I$, $W_o^a = H^T H$, where

$$H = \Sigma_o U_o^T U_c \Sigma_c. \quad (B-47)$$

The singular values of H are equal to those of W_o . This forces an input normalization.

Let us look at the matrix H in more detail. The continuous time grammians are related to their discrete-time matrix counterparts via

$$W_c^a = \lim_{h \rightarrow 0} \frac{1}{h} [M_{con}] [M_{con}]^T \quad (B-48)$$

$$W_o^a = \lim_{h \rightarrow 0} h [M_{obs}]^T [M_{obs}]$$

where h is sampling time

M_{con} is the discrete-time extended controllability matrix,

M_{obs} is the discrete-time extended observability matrix.

Let W_c , W_o have factorizations $U_c \Sigma_c U_c^T$ and $U_o \Sigma_o U_o^T$, respectively. Let $M_{con} = U_{con} \Sigma_{con} V_{con}^T$, and $M_{obs} = U_{obs} \Sigma_{obs} V_{obs}^T$. Moore shows that

$$U_c = \lim_{h \rightarrow 0} U_{con} \quad \Sigma_c = \lim_{h \rightarrow 0} \frac{1}{\sqrt{h}} \Sigma_{con} \quad (B-49)$$

$$U_o = \lim_{h \rightarrow 0} U_{obs} \quad \Sigma_o = \lim_{h \rightarrow 0} \frac{1}{\sqrt{h}} \Sigma_{obs}$$

$$\sigma_H = \lim_{h \rightarrow 0} \sigma_{HANKEL} \quad (B-50)$$

Equation (B-50) offers an interesting result. The infinite dimensional Hankel matrix $([M_{obs}]) [(M_{con})]$ is coordinate invariant. This result shows that essential information contained in the Hankel matrix is embodied in the H matrix. The singular values of the Hankel matrix give a measure of closeness to redundancy of state variables. Therefore, the singular value decomposition of Moore's H matrix

gives us pertinent information about the redundant state variables. With this H matrix, loosely we have injected controllability/observability properties into a single matrix.

Defining a new system related to the original system via

$$\underline{X} = T \underline{Z} \quad (B-51)$$

$$\text{where } T = U_O \Sigma_O^{-1} U_H \Sigma_H^{\frac{1}{2}} \quad (B-52)$$

$$H = U_H \Sigma_H V_H^T$$

produces the "internally balanced" state representation. This transformation forces $W_C = W_O = \Sigma_H^{\frac{1}{2}}$. Thus, the internally balanced system is balanced with respect to an "equal" weighting on controllability and observability properties.

Many desirable properties result from the internally balanced representation. Only one of which is the fact that the singular values of the H matrix give us an estimate of how low we can reduce the order of the system without severe loss of accuracy. This can be viewed most easily again from the geometrical point of view. As discussed previously, the H matrix contains information concerning the observability and controllability properties of the particular system. If we observe the singular value of H as

$$\sigma_1, \sigma_2, \dots, \sigma_k, \sigma_{k+1}, \dots, \sigma_n \quad (B-53)$$

where $\sigma_k \gg \sigma_{k+1}$, then since the singular values are the lengths of the axes of the hyperellipsoid, can be thought of as being associated with the major axes of the hyperellipsoid. Then, σ_{k+1} through σ_n can be thought of as the minor axes of the hyperellipsoid.

Therefore, if indeed $q_k \gg q_{k+1}$, we can view the axes corresponding to σ_{k+1} through σ_n as being degenerate. As mentioned previously, "degeneracy" corresponds to state variable "redundancy" for our coordinate invariant system. Since the H matrix embodies controllability/observability properties, to strip away the states corresponding to σ_{k+1} through σ_n is equivalent to stripping away the state redundancy of the system. Moreover, it is equivalent to stripping away the uncontrollable, unobservable subspace. This is analogous to the Kalman decomposition (Ref 25:20).

Many equations have been presented. However, the development presented by Moore in References 24 and 25 is much more extensive. The reader is encouraged to refer to the material for assistance in grasping the technique.

Section II will take a "step away" from the equations and present the rationale behind the development in a more clear fashion. Also, important properties that this "internally balanced" representation possess, not mentioned here, will be presented. This Appendix was designed to help the interested reader better understand the development of the Moore algorithm, as well as to base the results of Sections III and V upon.

APPENDIX C

USER'S MANUAL FOR MIMO

August 1979

USER'S MANUAL FOR MIMO

This user's manual was prepared by 2nd Lt James R. McClendon, USAF in partial fulfillment of his Masters Thesis at AFIT/ENG, Wright-Patterson AFB, Ohio, August 1979.

****	****	IIII	****	****	00000000
*****	*****	II	*****	*****	00 00
*****	*****	II	*****	*****	00 00
*****	*****	II	*****	*****	00 00
*****	*****	II	*****	*****	00 00
****	****	II	****	****	00 00
****	****	II	****	****	00 00
****	****	II	****	****	00 00
****	****	II	****	****	00 00
****	****	IIII	****	****	00000000

AN INTERACTIVE COMPUTER PROGRAM CONTAINING
SEVERAL OPTIONS USEFUL IN CHANGING STATE VARIABLE
COORDINATE SYSTEMS AND PROVIDING REDUCED ORDER
MODELS.

AIR FORCE INSTITUTE OF TECHNOLOGY
AUGUST 1979

OVERVIEW OF MIMO

MIMO is designed as a tool for the user interested in model order reduction and singular value analysis of multi-variable, linear, time-invariant systems.

MIMO offers a model order reduction algorithm that is highly accurate for both impulse and step response applications.

MIMO provides options to obtain the singular values and the closely related condition number with respect to inversion.

MIMO plots as well as lists the frequency response for a user specified reduced order model.

MIMO also obtains special coordinate systems. These coordinate systems include: the output normal system(states ordered with respect to observability), the output predictive system(where the observability matrix is the identity matrix), and the internally balanced system(which the model order reduction technique is based upon).

MIMO should result in a substantial time savings to the user interested in a reduced order model for his particular system, or interested in investigating the relative importance of states as they effect input/output properties of the system. This qualitative analysis of the states is based heavily upon concepts of singular value analysis(Ref 13,36).

This user's manual underlines everything the user should type in. (Some inputs are circled instead instead for clarity)

MIMO PRELIMINARIES

TO ACCESS MIMO

TYPE: ATTACH MIMO, ID=AFIT

MIMO

The previous two lines will attach the permanent file MIMO. The program will begin execution with the typing of the word MIMO.

PREPARATION

The user should prepare himself with the proper input before a session is started. If the user desires to use MIMO for any state-space related options, he needs the input (A,B,C) system at his disposal. The user will require this (A,B,C) for any state coordinate system option. MIMO does not require the D matrix if one exists for its work.

The user may obtain singular values, singular vectors, and estimate the condition number with respect to inversion without inputting the (A,B,C) system through Option 2 first.

MIMO has various options to offer which are listed and described in detail in the next section. Most options explain themselves. They all ask for the desired inputs. However, if at any time should a question arise, referral to this user's manual should answer it.

TABLE OF CONTENTS

ITEM	PAGE
Overview.....	i
Preliminaries.....	ii
Option 0--List all Options.....	1
Option 1--Stop and Update File MEMORE.....	1
Option 2--Moore Algorithm(state space input).....	2
Option 3--Discretize System(yield F,G, discrete time controllability, observability and Hankel matrices.....	6
Option 4--Plot and Optionally List Singular Values of Controllability, Observability, and Hankel matrices vs. sample time.....	7
Option 5--Obtain continuous time controllability and observability matrices.....	9
Option 6--Estimate the Condition Number of square matrix.....	9
Option 7--List Singular Values, Vectors for a matrix.	10
Option 8--Plot and List Frequency Response for balan- ce system.....	10
Option 9--Dispose Plots to AFIT Plotter.....	11
Option 10--Update file MEMORE.....	11
Option 11--Recover Information from MEMORE.....	12
Option 12--TOTAL Interface.....	12
Example of Option 12.....	13
Example of reducing order of balanced system in TOTAL	17
Option 13--Output Normal State Coordinate System.....	19
Option 14--Output Predictive State Coordinate System.	19
Option 15--Steady state Controllability, Observability Grammians.....	20
Plot file Options.....	21

DESCRIPTION OF OPTIONS

OPTION 0 List all options

TYPE '0' (ZERO) FOR A LIST OF OPTIONS
TYPE OPTION NUMBER >0

SIXTEEN OPTIONS ARE NOW PROVIDED

- OPTION 0: LIST OPTIONS
- OPTION 1: STOP
- OPTION 2: OBTAIN THE BALANCED (A', B', C') , GIVEN (A, B, C)
- OPTION 3: OBTAIN THE DISCRETE TIME F AND G MATRICES
AND THE CORRESPONDING DISCRETE TIME
CONTROLLABILITY, OBSERVABILITY AND
HANKEL MATRICES
- OPTION 4: PLOT SINGULAR VALUES OF DISCRETE
CONTROLLABILITY, OBSERVABILITY, AND
HANKEL MATRICES VS. SAMPLE TIME
- OPTION 5: OBTAIN THE CONTINUOUS TIME
CONTROLLABILITY AND OBSERVABILITY
MATRICES
- OPTION 6: ESTIMATE CONDITION NUMBER OF A MATRIX
- OPTION 7: OBTAIN THE SINGULAR VALUES FOR A GIVEN
MATRIX-----NEED NOT BE SQUARE
- OPTION 8: BODE PLOT OF USER SPECIFIED TRANSFER
FUNCTION (BALANCED SYSTEM) FOR GIVEN
ORDER SYSTEM
- OPTION 9: DISPOSE PLOTS TO PLOTTER
- OPTION 10: SAVE ALL INFORMATION IN FILE MEMORE
- OPTION 11: RECOVER INFO FROM LOCAL FILE--MEMORE
- OPTION 12: TOTAL INTERFACE
- OPTION 13: OBTAIN OUTPUT NORMAL REP.
OPTION 2 MUST BE EXECUTED FIRST
- OPTION 14: OBTAIN OUTPUT-PREDICTIVE REP.
- OPTION 15: OBTAIN THE CONTINUOUS TIME CONTROLLABILITY
AND OBSERVABILITY GRAMMIANS FOR THE INPUT
 (A, B, C) SYSTEM. (WARNING---THE A MATRIX MUST
HAVE ONLY NEGATIVE EIGENVALUES!!!)

OPTION 1 Stop

Terminates execution of MIMO. Saves all current information in local file MEMORE. If OPTION 12 has been executed, the TOTAL interface local file MEMAUX is created.

If the user has executed a plotting option, the PLOT file will be created.

This PLOT file will be empty if OPTION 9 has been executed after either of the plotting options(4or8) has been executed. This occurs because of the peculiarities of the software required for OPTION 9. Therefore, the PLOT file may be returned. The PLOT file will contain plots if OPTION 9 has not been executed after either of the plotting options(4or8) has been executed.

OPTION 2 Input (A,B,C) and optionally obtain the
internally balanced representation

(The maximum dimension of A,B, or C is 10X10)

a) Options 2b,3,4,5,6,7,8,9,10,11,12,13, and 14 use the (A,B,C) system input by this option. Input must occur by rows. The matrices are printed out for verification, and the opportunity to change mistakes is provided. You may then terminate OPTION 2, or obtain the internally balanced representation.

TYPE OPTION NUMBER 2

THIS PROGRAM INTERNALLY BALANCES AN (A,B,C)
REPRESENTATION YIELDING A (A',B',C')
REPRESENTATION.

ENTER ORDER OF AMAT 2
NOW ENTER THE AMAT AS FOLLOWS
PLEASE ENTER THE MATRIX BY ROWS

ROW(1)=0.1

ROW(2)=-5,-2

1 0. 1.0000E+00

2 -5.0000E+00 -2.0000E+00

DO YOU WISH TO CHANGE ANY ELEMENTS?(1=YES,2=NO) 2

ENTER THE COLUMN DIMENSION(S) OF INPUTS OF THE BMAT (1)
PLEASE ENTER THE MATRIX BY ROWS

ROW(1)=0

ROW(2)=1

1 0.

2 1.0000E+00

DO YOU WISH TO CHANGE ANY ELEMENTS? (1=YES, 2=NO) (2)

ENTER THE ROW DIMENSION(S) OF OUTPUTS OF THE CMAT

NOTE: IF SISO SYSTEM, CMAT=(CVEC) TRANSPOSED (1)

PLEASE ENTER THE MATRIX BY ROWS

ROW(1)=1,0

1 1.0000E+00 0.

DO YOU WISH TO CHANGE ANY ELEMENTS? (1=YES, 2=NO) (2)

b) If the balancing path is chosen, you must then choose whether the balanced system should better approximate; (1) the response to an impulse input, or (2) the response to a step input. This is not to say the impulse response balanced system reduced order model will not give excellent results to a step input or any other class of inputs. However, in order to obtain more accurate, lower order models to respond to a step input, the system should be balanced with respect to a step input.

If the step response balancing path is chosen, a D' matrix will be created. MIMO asks the user the desired order of his reduced order model. Once entered, MIMO will obtain the D' matrix which should be added directly to the original D matrix (if one did not exist--this D' matrix becomes the necessary D matrix to provide good results). Note however, this D' matrix is appropriate only for the reduced order model for which it is derived.

Do not attempt to add this D' matrix to any reduced order system other than the order specified by the user. The user should try different combinations of the impulse and step balancing paths to obtain the most accurate reduced order model for his desired input.

Reduction of order is accomplished by deleting the bottom most states of the resulting balanced representation.

i.e.

$$A_{\text{balanced}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$B_{\text{balanced}} = \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix} \quad C_b = [13 \ 14 \ 15]$$

$$A_{\text{balanced reduced}} = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$$

$$B_{\text{balanced reduced}} = \begin{bmatrix} 10 \\ 11 \end{bmatrix} \quad C_b = [13 \ 14]_r$$

The balancing algorithm requires; (1) the input system to be at least marginally stable, and (2) if the step balancing path is chosen, the input A matrix must not be singular.

A check for accuracy exists if and only if a SISO(single-input-single-output) system is input. This check consists of an absolute value symmetry occurring in the balanced A matrix. This check does not hold for MIMO(multi-input-multi-output) systems.

The user may print out the H_{INF} matrix used in the balancing algorithm. This matrix's singular values are printed out subsequently. These singular values are related to those of the discrete-time Hankel matrix via the following relation:

$$\lim_{t \rightarrow 0} \sigma_{\text{HANKEL}} = \sigma_{H_{\text{INF}}} ; \text{ where } t \text{ is sampling time and } \sigma \text{ represents singular values of the infinite order Hankel matrix}$$

The user can group these singular values into "large" and "small" groups. The number of "large" singular values can give an approximation to the lowest possible reduced order model that can be attained without severe loss of accuracy.

The B' and C' for output matrices should be used for time and frequency domain responses. The B' and C' matrices order the inputs and outputs with respect to their l_2 norm. This indicates the "influence" of the inputs, and the "responsiveness" of outputs.

DO YOU WISH TO CONTINUE TO OBTAIN THE
BALANCED REPRESENTATION? (1=YES, 2=NO) ①
DO YOU WANT THE BALANCED REPRESENTATION
TO BETTER APPROXIMATE-1. IMPULSE RESPONSE OF
FULL ORDER SYSTEM-OR 2. STEP RESPONSE OF FULL
ORDER SYSTEM? ENTER 1/2
DO YOU WISH TO SEE THE HINF MAT. (1=YES, 2=NO) ①
THE HINF MATRIX IS

1 -6.1559E-02 3.6045E-02

2 -3.1918E-03 -3.8742E-02
FOR HELP IN ESTIMATION ON LIMITS OF ORDER
REDUCTION POSSIBILITY, THIS SINGULAR VALUE
VECTOR IS PRINTED OUT. TO UTILIZE, GROUP ALL
'LARGE' SINGULAR VALUES. THIS NUMBER IS
GENERALLY THE SMALLEST ONE CAN REDUCE THE SYSTEM
WITHOUT SIGNIFICANT LOSS OF ACCURACY.

1 7.3851E-02

2 3.3851E-02

THE A' MATRIX IS

1 -1.5571E+00 2.0761E+00

2 -2.0761E+00 -4.4291E-01

THE B' MATRIX IS

1 -3.8721E-01

2 -1.0723E+00

THE C' MATRIX IS

1 -4.7957E-01 1.7317E-01

FOR OUTPUT THE BMAT IS

1 3.8721E-01

2 1.0723E+00
FOR OUTPUT THE CMAT IS

1 4.7957E-01 -1.7317E-01
ENTER DESIRED ORDER OF REDUCED SYSTEM (1)
THE D' MATRIX IF BALANCED SYSTEM IS
REDUCED TO 1 IS .

1 8.0742E-02

OPTION 3 Discretize (A,B,C) or (A',B',C') and obtain the
F, G, discrete-time controllability, observability,
and Hankel matrices

A truncated (50 term) power series expansion is used to obtain the F and G matrices for the (A,B,C) system, or the (A',B',C') system. Both systems are results of OPTION 2. The (F,G,C) matrices are used to obtain the discrete-time controllability, observability, and Hankel matrices.

TYPE OPTION NUMBER (3)

DO YOU WISH TO SUPPRESS THE PRINTOUTS? (1=YES, 2=NO) (2)

DO YOU WISH TO DISCRETIZE THE ORIGINAL (1) SYSTEM
OR THE BALANCED (2) SYSTEM? ENTER (1)
ENTER THE SAMPLING TIME (T) (4)

THE F MATRIX IS

1 7.0745E-01 2.4043E-01

2 -1.2021E+00 2.2659E-01

THE G MATRIX IS

1 5.8511E-02

2 2.4043E-01

THE DISCRETE TIME CONTROLLABILITY
MATRIX IS

1 5.8511E-02 9.9199E-02

2 2.4043E-01 -1.5860E-02
THE DISCRETE TIME OBSERVABILITY
MATRIX IS

1 1.0000E+00 0.

2 7.0745E-01 2.4043E-01
THE DISCRETE TIME HANKEL MATRIX IS

1 5.8511E-02 9.9199E-02

2 9.9199E-02 6.6365E-02

OPTION 4 Plot and optionally list singular values and
their ratio

The listing may be suppressed. The discrete-time controllability, observability, Hankel matrix, or all three matrices' singular values may be plotted and optionally listed. The user may plot any two(2) singular values and their ratio for each matrix versus sample time. The singular values are ordered from minimum to maximum. (Sig(1) is the minimum singular value, Sig(n) the maximum)

The user is asked to input the indices of the singular values he wishes to plot.(i.e. 3,5). If 3,5 is entered, singular value #3, singular value #5, and the ratio singular value #3/singular value #5 is plotted. The first indice entered by the user will subsequently be referred to as the "minimum" singular value. Similarly, the second indice entered by the user will be referred to as "maximum".

USES: The plot of the condition number of the Hankel matrix is a great aid in choice of sample time. The user should pick a sample time that minimizes this condition number. (Condition number = $\text{Sig}(\text{max})/\text{Sig}(\text{min})$). This will make the input (F,G,C) system more robust to perturbations in its elements if the sample time is chosen via this process.

The other available plots give the user an idea of the degree of control or the responsiveness of outputs to a given input that a particular system possess.

TYPE OPTION NUMBER ④

THE CONTROLLABILITY AND OBSERVABILITY MATRICES EACH HAVE 2 IMPORTANT SINGULAR VALUES. THEY ARE ORDERED FROM MINIMUM TO MAXIMUM IN MAGNITUDE. ENTER-- 1 TO PLOT CONTROLLABILITY, 2 TO PLOT TO PLOT OBSERVABILITY, 3 TO PLOT HANKEL, OR 4 TO PLOT ALL ⑤

YOU MAY PLOT 2 SINGULAR VALUES AND THEIR RATIO VS. SAMPLE TIME. I.E. $\text{SIG}(N1), \text{SIG}(N2), \text{SIG}(N1)/\text{SIG}(N2)$ ENTER THE 2 SINGULAR VALUE NUMBERS IN DESIRED ORDER FOR RATIO PLOT-- $\text{SIG}(I.D.1)/\text{SIG}(I.D.2)$ --(I.E. ENTER 2,5--WILL PLOT SINGULAR VALUES #2, AND #5 AND #2/#5) ENTER ① ⑥

THE 1ST I.D. WILL BE REFEFFED TO AS MIN, 2ND--MAX DO YOU WISH TO SUPPRESS THE PRINTOUT? (1=YES, 2=NO) ②

NOTE :THE FOLLOWING TITLE BLOCKS ARE PROVIDED FOR YOUR CONVENIENCE. THE PLOTS ARE ALREADY WELL-LABELED AS TO WHICH ONES THEY ARE. THE TITLE BLOCKS MAY BE USED FOR USER DEFINED LABELS--SUCH AS FIGURE NUMBERS ETC.

PLOT OF THE MAXIMUM SING. VAL. OF HANKEL MAT.

> (-----ENTER TITLE (50 CHARACTERS MAX)-----) <

> FIGURE 1 MINIMUM SINGULAR VALUE OF HANKEL MAT

FIGURE 1 MINIMUM SINGULAR VALUE OF HANKEL MAT

PLOT OF THE MINIMUM SING. VAL. OF HANKEL MAT.

> (-----ENTER TITLE (50 CHARACTERS MAX)-----) <

> FIGURE 2 MAXIMUM SINGULAR VALUE OF HANKEL MAT

FIGURE 2 MAXIMUM SINGULAR VALUE OF HANKEL MAT

PLOT OF THE CONDITION NUMBER OF HANKEL MAT.

> (-----ENTER TITLE (50 CHARACTERS MAX)-----) <

> FIGURE 3 RATIO OF SIG1/SIG2

FIGURE 3 RATIO OF SIG1/SIG2

NOTE: THE CALCULATION DELTA FOR THIS ALGORITHM IS THE INITIAL TIME. THEREFORE THE INITIAL TIME MUST

BE LESS THAN ONE. THE FINAL TIME MUST BE WITHIN
98 STEPS OF THE INITIAL TIME
PLEASE ENTER THE BEGINNING SAMPLING TIME
AND THE FINAL SAMPLING TIME 2.1.5
THE SINGULAR VALUES OF THE HANKEL MAT. FOR TIME .1 ARE >

1 2.7990E-03

2 2.6051E-02

OPTION 5 Obtain the continuous time controllability and observability matrices

Obtains the continuous time controllability and observability matrices for (A,B,C) system input by OPTION 2.

OPTION 6 Estimate condition number of a matrix

This option uses a package of highly efficient LINPACK (SIAM Publishing Co.) subroutines to estimate the condition number of a matrix. This option does not require relatively expensive singular value decompositions. It is useful when the condition number's estimated magnitude is desired. Several options exist. The user can obtain the estimate for A(original), A'(balanced, F(must be created by OPTION 3 first), or the H_{INF} matrix, a matrix input at this time, or the discrete-time Hankel matrix(must be created by OPTION 3 first). Warning---except for the Hankel matrix, all matrices must be square!

TYPE OPTION NUMBER 6

THIS OPTION ESTIMATES THE CONDITION NUMBER OF A SQUARE MATRIX. ENTER THE NUMBER CORRESPONDING TO THE DESIRED INPUT MATRIX

1-----AMAT(ORIGINAL)

2-----AMAT(BALANCED)

3-----F

4-----ENTER YOUR OWN

5-----MATRIX TO HELP EVALUATE ORDER REDUCTION

6-----HANKEL MAT. (MUST BE CREATED BY OPT. 3)

ENTER 1

THE ESTIMATED CONDITION NUMBER IS RCOND= 6.25

OPTION 7 Prints out singular values and optionally the
left and right singular vectors of input matrix

The user can again specify the same matrices as above in
OPTION 6, except for the H_{INF} matrix. This option does not
require the matrices to be square. The user may print out the
left singular vectors, the right singular vectors, both, and the
singular values.

TYPE OPTION NUMBER (7)

THIS OPTION OBTAINS THE SINGULAR VALUES OF A GIVEN
MATRIX--NEED NOT BE SQUARE. THE RIGHT SINGULAR
VECTORS ARE

ENTER DESIRED CORRESPONDING NUMBER

1-----AMAT (ORIGINAL)

1 -9.2388E-01 3.8268E-01

2-----AMAT (BALANCED)

2 -3.8268E-01 -9.2388E-01

3-----F

4-----ENTER YOUR OWN

THE NON-ZERO SINGULAR VALUES ARE:

5-----DISCRETE HANKEL MATRIX

ENTER (1)

1 5.3983E+00

DO YOU WISH TO PRINT OUT THE

SINGULAR VECTORS AS WELL?

2 9.2621E-01

1=NO, 2=LEFT S.V., 3=RIGHT S.V., 4=BOTH

ENTER (4)

THE LEFT SINGULAR

VECTORS ARE

1 -7.0889E-02 -9.9748E-01

2 9.9748E-01 -7.0889E-02

OPTION 8 Frequency response for given order balanced
system

This option uses the modal description of a system to cal-
culate the frequency response of a given order balanced sys-
tem. The magnitude(DB) and phase(DEG) plots will be crea-
ted. Optionally, a listing of 100 points can be obtained
starting with a user specified frequency(RAD/SEC).

TYPE OPTION NUMBER 8
FREQUENCY RESPONSE FOR BALANCED SYSTEM

THE BALANCED A MATRIX MUST BE CREATED BY OPTION 2 BEFORE THIS OPTION CAN BE EXECUTED!!!
THIS OPTION PLOTS THE FREQUENCY RESPONSE TO A GIVEN ORDER SYSTEM. ENTER DESIRED ORDER 1
SYSTEM HAS 1 INPUTS AND 1 OUTPUTS.
SELECT DESIRED INPUT-OUTPUT COMBINATION 1,3
PLEASE ENTER MAIN: -3 FOR .001, -2 FOR .01 ETC -1
1 - 1 INPUT-OUTPUT COMBINATION TRANSFER FUNCTION
FREQUENCY RESPONSE PLOT FOR 1 ORDER SYSTEM
MAGNITUDE PLOT (DB)
> (-----ENTER TITLE (50 CHARACTERS MAX)-----) <
> 1234466788
1234466788
PHASE PLOT (DEG)
> (-----ENTER TITLE (50 CHARACTERS MAX)-----) <
> PHASE PLOT
PHASE PLOT
DO YOU WISH TO SUPPRESS THE LISTING (1=YES, 2=NO) > 2
W MAG (DB) ANGLE (DEG)
.10E+00 -.1680E+02 -.1841E+03
.20 -.1686E+02 -.1881E+03
.30 -.1697E+02 -.1920E+03

OPTION 9 Dispose plots to AFIT plotter

This option requests a queue device for the plot file, closes it, and routes all accumulated plots to the AFIT plotter under your 2 letter user I.D. (when you login-the user I.D. assigned to you--can be obtained by typing "assets"). The plot file is then reopened. This option allows the user to route plots without ceasing execution of MIMO. The plot file must not exist if this program is reexecuted. If this option is not used, the standard operations with the plot file may be performed once execution of MIMO is terminated. See discussion on plot file options after this listing section.

OPTION 10 Update file MEMORE

This option updates the local file MEMORE with current, pertinent information, but does not cause termination of the program (this is automatically executed when OPTION 1 is chosen).

OPTION 11 Recover information from file MEMORE

This option recovers pertinent information from local file MEMORE created by OPTION 1 or OPTION 10. This option coupled with option 10 allows the user to cease execution of MIMO to do other things and come back and recover to his original point.

OPTION 12 TOTAL interface

This option interfaces MIMO to TOTAL. This saves the user from typing in large matrices into TOTAL to obtain responses. The option explains itself. The following list is the key to the interface:

MIMO	TOTAL
AMAT(original)	UMAT
BMAT(original)	VMAT
CMAT(original)	WMAT
AMAT(balanced)	XMAT
BMAT(balanced-output)	YMAT
CMAT(balanced-output)	ZMAT

These matrices are stored in a file called MEMAUX. This file is created by this option and will exist at termination of this program. This option (12) should be executed after the balanced system is obtained for options 13 and 14 will overwrite the original balanced system. The user should note that TOTAL will not alter MIMO in any way. In other words the interface is one-way. No TOTAL obtained matrices may be interfaced back into MIMO.

TYPE OPTION NUMBER ⑫

(A,B,C) AND (A',B',C') HAVE BEEN STORED IN SUCH A WAY THAT TOTAL CAN DIRECTLY ACCESS THEM. THROUGH USE OF THE COPY COMMAND UNDER TOTAL FULL FLEXIBILITY IS EXERCISED. THE FOLLOWING LIST IS THE KEY

MIND	STORED IN TOTAL
A ORIGINAL	UMAT
B ORIGINAL	VMAT
C ORIGINAL	WMAT
A BALANCED	XMAT
B BALANCED-OUTPUT	YMAT
C BALANCED-OUTPUT	ZMAT

USING TOTAL'S OPTION 25 THE TRANSFER FUNCTION MAY BE OBTAINED. WHEN MIND IS TERMINATED THE ABOVE INFORMATION IS STORED IN MASS STORAGE FORMAT ON THE LIKEWISE TOTAL NAMED FILE-----MEMAUX---THIS FILE MUST EXIST PRIOR TO TRYING TO INTERFACE WITH TOTAL!!!!!!

THE FOLLOWING PROCEDURE SHOULD BE UTILIZED.

FIRST COPY (UMAT,VMAT,WMAT) OF (XMAT,YMAT,ZMAT) INTO (AMAT,EMAT,CMAT). THEN TYPE OPTION 18 IN TOTAL. WHEN ASKED FOR THE NUMBER OF STATES--ENTER THE FULL NUMBER IF USING THE ORIGINAL SYSTEM. ENTER THE DESIRED ORDER IF USING THE BALANCE SYSTEM. THEN FOR EACH ELEMENT, TYPE A * (ASTERIK) WHICH MEANS TOTAL WILL KEEP THAT LOCATION THE SAME. THEN ENTER NO FOR THE DMAT AND KMAT IF APPROPRIATE. THEN OPTION 25 CAN BE EXECUTED.

*****EXAMPLE OF INTERFACE WITH TOTAL AND THE USE OF TOTAL*****

TYPE OPTION NUMBER ⑪

ALL DATA UPDATED IN LOCAL FILE MEMDPE

STOP END OF EXECUTION--ALL PLOT(S)

IN FILE PLOT

.589 CP SECONDS EXECUTION TIME

FILE QUOTA EXCEEDED

..RETURN,LGO,MEMDPE,PLOT

..TOTAL

WELCOME TO TOTAL--VERSION 1.4

TYPE HELP FOR INTPO, TYPE 99 FOR NEW FEATURES BULLETIN

OPTION > UMAT,VMAT,WMAT,XMAT,YMAT,ZMAT

COL >	1	2	3
ROW			
1	0.	1.000	0.
2	0.	0.	1.000
3	-1.000	-2.000	-2.000

COL >	1	2
FCN		
1	1.000	0.
2	0.	1.000
3	2.000	3.500

COL >	1	2	3
ROW			
1	1.000	2.500	1.000

COL >	1	2	3
ROW			
1	-.1195	-.1735	-.9220
2	-.1492	-1.171	-.3290
3	.9315	.3384	-.7098

COL >	1	2
ROW		
1	.1584	.8306
2	.4128	.4956
3	1.217	2.514

COL >	1	2	3
ROW			
1	-.8456	-.6450	2.793

OPTION > COPY,UMAT,AMAT,COPY,VMAT,BMAT,COPY,DMAT,CHAT

COPY COMPLETE

COPY COMPLETE

COPY COMPLETE

OPTION > (18)

IS THE SYSTEM (1) CONTINUOUS OR (2) DISCRETE? > (1)

THE EQUATIONS YOU ARE ABOUT TO INPUT HAVE THE FORM:

$$\dot{X}(T) = [A]X(T) + [B]U(T)$$

$$Y(T) = [C]X(T) + [D]U(T)$$

WHERE $U(T) = \text{GAIN} \cdot R(T) - [K]X(T)$

AND X IS A VECTOR OF N STATE VARIABLES

U IS A VECTOR OF M INPUTS

Y IS A VECTOR OF L OUTPUTS

ENTER NO. OF STATES, INPUTS, OUTPUTS > 3,2,1

ENTER AMAT WITH 3 ROWS AND 3 COLUMNS.

ENTER 3 ELEMENTS PER ROW:

ROW 1 > ♦♦♦♦
ROW 2 > ♦♦♦♦
ROW 3 > ♦♦♦♦

COL >	1	2	3
ROW			
1	0.	1.000	0.
2	0.	0.	1.000
3	-1.000	-2.000	-2.000

ENTER BMAT WITH 3 ROWS AND 2 COLUMNS.

ENTER 3 ELEMENTS PER COLUMN:

COLUMN 1 > ♦♦♦♦
COLUMN 2 > ♦♦♦♦

COL >	1	2
ROW		
1	1.000	0.
2	0.	1.000
3	2.000	3.500

ENTER CMAT WITH 1 ROWS AND 3 COLUMNS.

ENTER 3 ELEMENTS PER ROW:

ROW 1 > ♦♦♦♦

COL >	1	2	3
ROW			
1	1.000	2.500	1.000

IS THERE A DIRECT-TRANSMISSION (D MATRIX)--YES OR NO? > (NO)

DMAT SET TO 1 BY 2 ZERO MATRIX (OPTION 78)

IS THERE A STATE-VARIABLE FEEDBACK MATRIX--YES OR NO? > (NO)

KMAT HAS BEEN SET TO 2 BY 3 ZERO MATRIX

THE STATE-SPACE REPRESENTATION IS COMPLETE.

OPTION > (25)

SYSTEM HAS 2 INPUT(S) AND 1 OUTPUT(S).

SELECT WHICH TRANSFER FUNCTION IS DESIRED:

ENTER WHICH INPUT, WHICH OUTPUT > (1,1)

GTF(S) AND HTF(S) CALCULATED FROM STATE-SPACE

TYPE: GTF OR HTF FOR RESULTS

OPTION > GTF,HTF

FORWARD-LOOP TRANSFER FUNCTION

$$GK = (GNK/GDK) = 3.000$$

GTF(S) NUMERATOR

I	GNPOLY(I)			GZERO(I)	
1	(3.000)	S++ 2	(-.2929)	+ JK	0.)
2	(6.000)	S++ 1	(-1.707)	+ JK	0.)
3	(1.500)				

GNK= 3.000

GTF(S) DENOMINATOR

I	GDPOLY(I)			GPOLE(I)	
1	(1.000)	S++ 3	(-.5000)	+ JK	.8660)
2	(2.000)	S++ 2	(-.5000)	+ JK	-.8660)
3	(2.000)	S++ 1	(-1.000)	+ JK	0.)
4	(1.000)				

GDK= 1.000

FEEDBACK-LOOP TRANSFER FUNCTION

$$HK = (HNK/HDK) = 0.$$

HTF(S) NUMERATOR

I	HNPOLY(I)			HZERO(I)	
1	(0.)				

HNK= 0.

HTF(S) DENOMINATOR

I	HDPOLY(I)			HPOLE(I)	
1	(3.000)	S++ 2	(-.2929)	+ JK	0.)
2	(6.000)	S++ 1	(-1.707)	+ JK	0.)
3	(1.500)				

HDK= 3.000

OPTION > COPY,GTF,DLTF
COPY COMPLETE

OPTION > CLOSED,OFF,39
OPEN-LOOP MODE SET

TYPES OF INPUT: (1) IMPULSE, (2) STEP, (3) RAMP,
(4) PULSE, (5) SINE...SELECT ONE > (2)
ENTER STEP INPUT MAGNITUDE (1)

OPTION > 34
CONTINUOUS TIME RESPONSE FOR DLTF(S)
WITH STEP INPUT OF STRENGTH = .1

ENTER INITIAL TIME, FINAL TIME > 0,10
> [-----ENTER TITLE (50 CHARACTERS MAX)-----] <
> TIME RESPONSE OF 3RD ORDER ORIGINAL SYSTEM
TIME RESPONSE OF 3RD ORDER ORIGINAL SYSTEM

*****EXAMPLE OF REDUCING THE ORDER OF THE BALANCED SYSTEM*****

OPTION > COPY, XMAT, AMAT, COPY, YMAT, BMAT, COPY, ZMAT, CMAT
 COPY COMPLETE
 COPY COMPLETE
 COPY COMPLETE

OPTION > (18)
 IS THE SYSTEM (1) CONTINUOUS OR (2) DISCRETE? > (1)

THE EQUATIONS YOU ARE ABOUT TO INPUT HAVE THE FORM:

$$\dot{X}(T) = [A]X(T) + [B]U(T)$$

$$Y(T) = [C]X(T) + [D]U(T)$$

WHERE $U(T) = \text{GAIN} \times R(T) - [K]X(T)$

AND X IS A VECTOR OF N STATE VARIABLES
 U IS A VECTOR OF M INPUTS
 Y IS A VECTOR OF L OUTPUTS

ENTER NO. OF STATES, INPUTS, OUTPUTS > 2,2,1
 ENTER AMAT WITH 2 ROWS AND 2 COLUMNS.

ENTER 2 ELEMENTS PER ROW:

ROW 1 > ♦♦♦

ROW 2 > ♦♦♦

COL > 1 2
 ROW
 1 -0.1195 -0.1735
 2 -0.1492 -1.171

ENTER BMAT WITH 2 ROWS AND 2 COLUMNS.

ENTER 2 ELEMENTS PER ROW:

ROW 1 > ♦♦♦

ROW 2 > ♦♦♦

COL > 1 2
 ROW
 1 .1584 .8306
 2 .4128 .4956

ENTER CMAT WITH 1 ROWS AND 2 COLUMNS.

ENTER 2 ELEMENTS PER ROW:

ROW 1 > ♦♦♦

COL > 1 2
 ROW
 1 -.8456 -.6450

IS THERE A DIRECT-TRANSMISSION (D MATRIX)--YES OR NO? > (NO)
 DMAT SET TO 1 BY 2 ZERO MATRIX (OPTION 78)
 IS THERE A STATE-VARIABLE FEEDBACK MATRIX--YES OR NO? > (NO)
 KMAT HAS BEEN SET TO 2 BY 2 ZERO MATRIX
 THE STATE-SPACE REPRESENTATION IS COMPLETE.

OPTION > (25)
 SYSTEM HAS 2 INPUT(S) AND 1 OUTPUT(S).
 SELECT WHICH TRANSFER FUNCTION IS DESIRED:
 ENTER WHICH INPUT, WHICH OUTPUT > 1,1
 GTF(S) AND HTF(S) CALCULATED FROM STATE-SPACE
 TYPE: GTF OR HTF FOR RESULTS

OPTION > GTF
 FORWARD-LOOP TRANSFER FUNCTION

$$GK = (GNK/GDK) = -.4002$$

GTF(S) NUMERATOR

1	GNPOLY(I)		GZERO(I)	
1	(-.4002)	S** 1	(-.2819)	+ J(0.)
2	(-.1128)		GNK=	-.4002

GTF(S) DENOMINATOR

1	GDPOLY(I)		GPOLE(I)	
1	(1.000)	S** 2	(-.9545E-01)	+ J(0.)
2	(1.290)	S** 1	(-1.195)	+ J(0.)
3	(.1140)		GDK=	1.000

OPTION > STOP
 LOCAL FILE--PLOT--CONTAINS CALCOMP PLOT(S)
 ALL INFO IN TOTAL HAS BEEN SAVED IN LOCAL FILE--MEMORY.

STOP

1.408 CP SECONDS EXECUTION TIME

FILE QUOTA EXCEEDED

..FILES

--LOCAL FILES--

\$INPUT	\$OUTPUT	WIND	♦PERM	\$INPUT
\$OUTPUT	♦JIMLIB	♦IMSL	♦TOTAL	MEMAUX
PLOT	MEMORY			

FILE QUOTA EXCEEDED

OPTION 13 Output normal state coordinate system

This option uses the (A', B', C') system obtained from Option 2 to result in a new (A^*, B^*, C^*) system which is balanced with respect to observability conditions only.

TYPE OPTION NUMBER (13)
THIS OPTION FINDS THE OUTPUT-NORMAL REP.
IT ALSO FINDS THE CORRESPONDING (F, G, C) SYSTEM
AT THE USER'S OPTION-THE OBS. MAT. MAY BE FORMED.
THE BEAR MATRIX IS

1 -1.4082E+00 1.3232E+00

2 -3.1489E+00 -5.9175E-01
THE BEAR MATRIX IS

1 -1.3460E+00

2 0.
THE CEAR MATRIX IS

1 3.7146E+00 4.9151E+00
PLEASE ENTER THE SAMPLING TIME (.3)
THE F MATRIX IS

1 5.2604E-01 2.7674E-01

2 -6.5860E-01 6.9681E-01
THE G MATRIX IS

1 -3.1011E-01

2 1.5210E-01
DO YOU WISH TO OBTAIN THE DISCRETE OBS. MAT (1=YES, 2=NO) (2)

OPTION 14 Output predictive state coordinate system

This option creates an (A^*, B^*, C^*) system such that the resulting observability matrix is indeed the identity matrix!

TYPE OPTION NUMBER (14)
THIS OPTION FINDS THE OUTPUT PREDICTIVE REP.
ENTER SAMPLING TIME 0.34.
THE F' MATRIX IS

1 4.8453E-01 3.7659E-01

2 3.7659E-01 2.9270E-01
THE G' MATRIX IS

1 4.4554E-02

2 3.4629E-02
THE C' MATRIX IS

1 6.2341E-01 4.8453E-01
TYPE OPTION NUMBER (1)

For more information on this system the user should refer to
Major J. Gary Reid's EE 5.10 class notes.

OPTION 15 Obtain the controllability and observability
grammians

This option obtains the controllability and observability grammians for the (A,B,C) system input in OPTION 2.

WARNING-----The A matrix must contain only negative eigenvalues.

TYPE OPTION NUMBER (15)

THE CONTROLLABILITY GRAMMIAN IS

1 4.9999E-02 6.9574E-16

2 6.9574E-16 2.5000E-01

THE OBSERVABILITY GRAMMIAN IS

1 4.4999E-01 9.9998E-02

2 9.9998E-02 4.9999E-02

PLOT FILE OPTIONS

A. OPTION 9 (this program)

- 1) Execute MIMO---create plots
- 2) Use OPTION 9---automatically routes plots to the AFIT terminal under your 2 letter user I.D. (see explanation of OPTION 9).

B. Standard Route (use if after C., or if desired route to other building besides AFIT (building 640--BB))

- 1) Execute MIMO---create plots
- 2) Terminate MIMO (OPTION 1) without executing OPTION 9 for the plots to be routed
- 3) Type: Route Plot, TID=BB, ST=CSB, DC=PT (where BB could be any other terminal I.D.)

C. To use CCPREV to preview plots (WARNING--user must be at a TEKTRONICS terminal)

- 1) Execute MIMO---create plots
- 2) Terminate MIMO without executing OPTION 9 for the plots to be previewed
- 3) Type: ATTACH CCPREV. ID=AFIT
CCPREV, PLOT
WINDOW,0,0,50,50,N,R,E,P,999
- 4) Plots will be small. However, if all plots are not on screen use the MOVE command, i.e. Type:
MOVE,delta-x,delta-y
where delta-x, and delta-y are in screen inches. For example, to move the entire string of plots to the left 2 screen inches, and up 3 screen inches, Type:
MOVE,-2,3.

5) Then Type:

R,E,P,999

This will replot the string of plots. Repeat steps 4 and 5 until plot string is on the screen (if impossible refer to CCPREV users manual).

6) Type:

SAVE,1

7) Type:

CR

This will display a vertical, and a horizontal line on the screen. Using the white rotating dials located on the terminal, the lines should be moved to the bottom left hand corner of the plot to be previewed. Type any letter on the keyboard (if the user is on the 1200 Baud 4014 terminal you must also type a carriage return!). The lines will momentarily disappear, and then reappear. Move the lines to the upper, right hand corner of the plot to be previewed. Again, type any character on the keyboard. This defines a new window.

8) Type:

N,R,E,P,999

This will display the delineated plot over the entire screen.

9) After examination of this plot, and if another of the plot string is to be previewed, Type:

USE,1

10) Now, repeat steps 7-9 for a different plot.

11) When you are finished, Type:

END

12) You may then return the plot file, or execute sequences B or E.

E. To catalog the plot file as a permanent file

1) Execute MIMO

2) Terminate MIMO without executing OPTION 9 for the string of plots desired to be stored.

3) Type:

REQUEST,NAME,*PF

(where NAME is any character string but "plot" or a current local file name)

4) Type:

COPY,PLOT,NAME

CATALOG,NAME,PLOTFIL

(where "plotfil" is any name you desire)

5) This is useful if you are not on a TEKTRONICS terminal at the time of execution of MIMO, and later desire to preview those plots with CCPREV. (if that is the case--you merely attach "procfil" as a local file later and then Type:

CCPREV,XXXXX

(whereXXXXXis the local file name you attached "procfil"under)

APPENDIX D

LISTING OF MIMO

August 1979

LISTING OF MIMO

This Appendix contains the source listing for MIMO. The sequence numbers are on the listing to assist the user in using the program.


```

966?  RMCC(I0,K0)=0.
      SIVS(I0,I)=0.
      WRKL(I0,I0)=0.
      CONTINUE
      DO 967 I0=1,100
      TRAT(I0)=0.
      HMLH(I0)=0.
      MCHN(I0)=0.
      HMAX(I0)=0.
      SC1AX(I0)=0.
      S01AX(I0)=0.
      S01I4(I0)=0.
      C01DC(I0)=0.
      C01DC(I0)=0.
      SC1IN(I0)=0.
      CONTINUE
967?  N1=N2=N3=N4=NC=NC=MB=NB=0
      MF=J
      NCIM=10
      N01M1=11
      K1=5
      K0UT=5
      KPIHOM=7
      SUP=.FALSE.
      CALL DATE(AAA)
      CALL CLOCK(AA)
      WRITE(6,102) AAA,AA
      FORMAT(1X,"TIME IS EXECUTING ON ",A10," AT ",A10)
      PRINT," "
      PRINT,"WELCOME TO M140-----VERSION 1"
      PRINT,"IF ANY OPTION UTILIZING (P,R,2) IS DESIRED:INPUT MUST"
      PRINT,"OCCUR THROUGH OPTION 2. ALL OTHER OPTIONS USE THE INPUT"
      PRINT,"MAT,ICLS FOR THEIR WORK. MATRIX SIZES ARE A,3,C--10X10"
      PRINT,"THE CONTROLLABILITY AND OBSERVABILITY OPTIONS CAN HANDLE"
      PRINT,"3 INPUTS AND 3 OUTPUTS IF A ILSIS SYSTEM IS INPUT. IN"
      PRINT,"OTHER WORDS THE CONTROLLABILITY MATRIX IS LIMITED TO A"
      PRINT,"10X30, WHILE THE OBSERVABILITY MATRIX IS LIMITED TO A"
      PRINT,"30X10 MATRIX."
      100
      C
      C

```

```

000900
000910
000920
000930
000940
000950
000960
000970
000980
000990
001000
001010
001020
001030
001040
001050
001060
001070
001080
001090
001100
001110
001120
001130
001140
001150
001160
001170
001180
001190
001200
001210
001220
001230
001240
001250
001260
001270
001280
001290

```

[illegible]

```

PRINT,"
PRINT,"
PRINT,"OPTION 5:
PRINT,"
PRINT,"
PRINT,"OPTION 5:
PRINT,"OPTION 7:
PRINT,"
PRINT,"OPTION 8:
PRINT,"
PRINT,"
PRINT,"OPTION 9:
PRINT,"OPTION 10:
PRINT,"OPTION 11:
PRINT,"OPTION 12:
PRINT,"OPTION 13:
PRINT,"
PRINT,"OPTION 14:
PRINT,"OPTION 15:
PRINT,"
PRINT,"
PRINT,"TYPE OPTION NUMBER > "
READ,LOPT
IF(LOPT.EQ.1)GO TO 5
IF(LOPT.EQ.2)NSTMT=1
IF(LOPT.EQ.3)NSTMT=20
IF(LOPT.EQ.4)NSTMT=25
IF(LOPT.EQ.5)NSTMT=5
IF(LOPT.EQ.6)NSTMT=9
IF(LOPT.EQ.7)NSTMT=7
IF(LOPT.EQ.8)NSTMT=8
IF(LOPT.EQ.9)NSTMT=9
IF(LOPT.EQ.10)NSTMT=10
IF(LOPT.EQ.11)NSTMT=11
IF(LOPT.EQ.12)NSTMT=12
IF(LOPT.EQ.13)NSTMT=13
IF(LOPT.EQ.14)NSTMT=14
IF(LOPT.EQ.15)NSTMT=15
CONTROLLABILITY,OBSERVABILITY,AND"
LANKEL MATRICES VS. SAMPLE TIME"
OBTAIN THE CONTINUOUS TIME"
CONTROLLABILITY AND OBSERVABILITY"
MATRICES"
ESTIMATE CONDITION NUMBER OF A MATRIX"
OBTAIN THE SINGULAR VALUES FOR A GIVEN"
MATRIX-----NEED NOT BE SQUARE"
CODE PLOT OF USER SPECIFIED TRANSFER"
FUNCTION(BALANCED SYSTEM) FOR GIVEN"
DESIGN SYSTEM"
DISPOSE PLOTS TO PLOTTER"
SAVE ALL INFORMATION IN FILE MEMORE"
RECOVER INFO FROM LOCAL FILE--MEMORE"
TOTAL INTERFAC"
OBTAIN OUTPUT NORMAL RLP"
OPTION 2 MUST BE EXECUTED FIRST"
OBTAIN OUTPUT-PREDICTIVE REP."
OBTAIN THE CONTINUOUS TIME CONTROLLABILITY"
AND OBSERVABILITY GRAPHS FOR THE INPUT "
(A,B,C) SYSTEM. (WARNING---THE A MATRIX MUST"
HAVE ONLY NEGATIVE EIGENVALUES!!)"
001700
001710
001720
001730
001740
001750
001760
001770
001780
001790
001800
001810
001820
001830
001840
001850
001860
001870
001880
001890
001900
001910
001920
001930
001940
001950
001960
001970
001980
001990
002000
002010
002020
002030
002040
002050
002060
002070
002080
002090

```



```

002510
002510
002520
002530
002540
002550
002560
002570
002580
002590
002600
002610
002620
002630
002640
002650
002660
002670
002680
002690
002700
002710
002720
002730
002740
002750
002760
002770
002780
002790
002800
002810
002820
002830
002840
002850
002860
002870
002880
002890

C READ MATRICES FROM USER
PRINT,"HOW ENTER THE BMAT AS FOLLOWS"
CALL READM(M,N,A)
ALLON USER TO CHANGE ELEMENTS IF DESIRED
CALL PRINTM(M,N,A)
CALL MISTAKE(M,N,A)
PRINT,"ENTER THE COLUMN DIMENSION(# OF INPUTS) OF THE BMAT > "
READ,NB
CALL READM(M,NB,B)
CALL PRINTM(M,NB,B)
CALL MISTAKE(M,NB,B)
PRINT,"ENTER THE ROW DIMENSION(# OF OUTPUTS) OF THE CMAT"
PRINT,"NOTE: IF SIS) SYSTEM, CMAT=(CVEC)TRANPOSED > "
READ,NC
CALL READM(MC,NC,C)
CALL PRINTM(MC,NC,C)
CALL MISTAKE(MC,NC,C)
PRINT,"DO YOU WISH TO CONTINUE TO OBTAIN THE"
PRINT,"BALANCED REPRESENTATION? (1=YES,2=NO) > "
READ,I14
IF(I14=2)GO TO 999
PRINT,"DO YOU WANT THE BALANCED REPRESENTATION"
PRINT,"TO GET AN APPROXIMATE-1.IMPULSE RESPONSE OF"
PRINT,"FULL ORDER SYSTEM-OR 2.STEP RESPONSE OF FULL"
PRINT,"ORDER SYSTEM? ENTER > "
READ,I14
IF(I14=2)GO TO 3
DO I=1,1
DO J=1,N
WAVEG(I,J)=A(1,J)
CONTINUE
CALL CINTIF(WKAREA,M,10,WKAREA,1,MINMED,IER)
CALL MULT(WKAREA,N,SBAR,N,M,NB)
DO I=1,NB
DO J=1,NB
SBAR(I,J,I)=FBAR(I,J)
CONTINUE
GO TO 573
DO I=1,M,NB,1
DO J=1,NB,1

```

```

10      BTAN(J,I)=P(I,J)
11      CONTINUE
12      DO 20 I=1,M,1
13      DO 21 J=1,N,1
14      ATAN(J,I)=A(I,J)
15      CONTINUE
16      DO 30 I=1,M,1
17      DO 31 J=1,N,1
18      CTAN(J,I)=C(I,J)
19      CONTINUE
20      IF (I.EQ.1) CALL MMULT (3,PTAN,3PROC,M8,N8,M8)
21      IF (I.EQ.2) CALL MMULT (39R,3TRAN,3PROC,M8,N8,M8)
22      CALL MLEQN(AI,AN,TPC),WQUAR,.01)
23      CALL MMULT (CTAN,C3PROC,AC,MC,MC)
24      CALL MLEQN(C4,C3PROC),WQUAR2,.01)
25      CALL MMULT (WQUAR2,CTRI,WINMC,MC,MC,MC)
26      CALL MMULT (WINMC,WINMC,MC,MC,MC)
27      CALL MMULT (WQUAR2,WINMC,MC,MC,MC)
28      CALL MMULT (WINMC,WINMC,MC,MC,MC)
29      CALL MMULT (WINMC,WINMC,MC,MC,MC)
30      CALL MMULT (WINMC,WINMC,MC,MC,MC)
31      CALL MMULT (WINMC,WINMC,MC,MC,MC)
32      CALL MMULT (WINMC,WINMC,MC,MC,MC)
33      CALL MMULT (WINMC,WINMC,MC,MC,MC)
34      CALL MMULT (WINMC,WINMC,MC,MC,MC)
35      CALL MMULT (WINMC,WINMC,MC,MC,MC)
36      CALL MMULT (WINMC,WINMC,MC,MC,MC)
37      CALL MMULT (WINMC,WINMC,MC,MC,MC)
38      CALL MMULT (WINMC,WINMC,MC,MC,MC)
39      CALL MMULT (WINMC,WINMC,MC,MC,MC)
40      CALL MMULT (WINMC,WINMC,MC,MC,MC)
41      CALL MMULT (WINMC,WINMC,MC,MC,MC)
42      CALL MMULT (WINMC,WINMC,MC,MC,MC)
43      CALL MMULT (WINMC,WINMC,MC,MC,MC)
44      CALL MMULT (WINMC,WINMC,MC,MC,MC)
45      CALL MMULT (WINMC,WINMC,MC,MC,MC)
46      CALL MMULT (WINMC,WINMC,MC,MC,MC)
47      CALL MMULT (WINMC,WINMC,MC,MC,MC)
48      CALL MMULT (WINMC,WINMC,MC,MC,MC)
49      CALL MMULT (WINMC,WINMC,MC,MC,MC)
50      CALL MMULT (WINMC,WINMC,MC,MC,MC)
51      CALL MMULT (WINMC,WINMC,MC,MC,MC)
52      CALL MMULT (WINMC,WINMC,MC,MC,MC)
53      CALL MMULT (WINMC,WINMC,MC,MC,MC)
54      CALL MMULT (WINMC,WINMC,MC,MC,MC)
55      CALL MMULT (WINMC,WINMC,MC,MC,MC)
56      CALL MMULT (WINMC,WINMC,MC,MC,MC)
57      CALL MMULT (WINMC,WINMC,MC,MC,MC)
58      CALL MMULT (WINMC,WINMC,MC,MC,MC)
59      CALL MMULT (WINMC,WINMC,MC,MC,MC)
60      CALL MMULT (WINMC,WINMC,MC,MC,MC)
61      CALL MMULT (WINMC,WINMC,MC,MC,MC)
62      CALL MMULT (WINMC,WINMC,MC,MC,MC)
63      CALL MMULT (WINMC,WINMC,MC,MC,MC)
64      CALL MMULT (WINMC,WINMC,MC,MC,MC)
65      CALL MMULT (WINMC,WINMC,MC,MC,MC)
66      CALL MMULT (WINMC,WINMC,MC,MC,MC)
67      CALL MMULT (WINMC,WINMC,MC,MC,MC)
68      CALL MMULT (WINMC,WINMC,MC,MC,MC)
69      CALL MMULT (WINMC,WINMC,MC,MC,MC)
70      CALL MMULT (WINMC,WINMC,MC,MC,MC)
71      CALL MMULT (WINMC,WINMC,MC,MC,MC)
72      CALL MMULT (WINMC,WINMC,MC,MC,MC)
73      CALL MMULT (WINMC,WINMC,MC,MC,MC)
74      CALL MMULT (WINMC,WINMC,MC,MC,MC)
75      CALL MMULT (WINMC,WINMC,MC,MC,MC)
76      CALL MMULT (WINMC,WINMC,MC,MC,MC)
77      CALL MMULT (WINMC,WINMC,MC,MC,MC)
78      CALL MMULT (WINMC,WINMC,MC,MC,MC)
79      CALL MMULT (WINMC,WINMC,MC,MC,MC)
80      CALL MMULT (WINMC,WINMC,MC,MC,MC)
81      CALL MMULT (WINMC,WINMC,MC,MC,MC)
82      CALL MMULT (WINMC,WINMC,MC,MC,MC)
83      CALL MMULT (WINMC,WINMC,MC,MC,MC)
84      CALL MMULT (WINMC,WINMC,MC,MC,MC)
85      CALL MMULT (WINMC,WINMC,MC,MC,MC)
86      CALL MMULT (WINMC,WINMC,MC,MC,MC)
87      CALL MMULT (WINMC,WINMC,MC,MC,MC)
88      CALL MMULT (WINMC,WINMC,MC,MC,MC)
89      CALL MMULT (WINMC,WINMC,MC,MC,MC)
90      CALL MMULT (WINMC,WINMC,MC,MC,MC)
91      CALL MMULT (WINMC,WINMC,MC,MC,MC)
92      CALL MMULT (WINMC,WINMC,MC,MC,MC)
93      CALL MMULT (WINMC,WINMC,MC,MC,MC)
94      CALL MMULT (WINMC,WINMC,MC,MC,MC)
95      CALL MMULT (WINMC,WINMC,MC,MC,MC)
96      CALL MMULT (WINMC,WINMC,MC,MC,MC)
97      CALL MMULT (WINMC,WINMC,MC,MC,MC)
98      CALL MMULT (WINMC,WINMC,MC,MC,MC)
99      CALL MMULT (WINMC,WINMC,MC,MC,MC)
100     CALL MMULT (WINMC,WINMC,MC,MC,MC)

```

```

66      SIGU(I,1)=SQRT(SIGU(I,1))
        SIGC(I,1)=SQRT(SIGC(I,1))
        CONTINUE
        DO 73 I=1,M,1
            DO 74 J=1,N,1
                SIGU2(I,J)=.000
                SIGC2(I,J)=.000
                SIGU(I,1)=SIGU(I,1)
                SIGC(I,1)=SIGC(I,1)
            CONTINUE
            CALL MULT(C1,G02,U01R1H,MIN2D,M,M,M)
            CALL MULT(MIN2D,U0,4KAREA,M,M,M)
            CALL MULT(MKAREA,SIG22,HINF,M,M,M)
            PRINT,"DO YOU WISH TO SEE THE MINF MAT.(1=YES,2=NO) > "
            READ,IVES
            IF(IVES.EQ.1)PRINT,"THE 4INF MATRIX IS "
            IF(IVES.EQ.2)CALL PRINTM(M,M,HINF)
            CALL LOVAL(MHINF,M,M,1,1,2,MKAREA,SIGH,UM,VH)
            DO 75 I=1,N,1
                DO 76 J=1,M,1
                    VHM2(I,J)=VH(I,J)
                CONTINUE
                DO 80 I=1,N,1
                    MKAREF(I,1)=SIGH(I,1)
                    SIGH(I,1)=SQRT(SIGH(I,1))
                CONTINUE
                DO 81 I=1,N,1
                    SIGH2(I,J)=.000
                CONTINUE
                SIGH2(I,1)=SIGH(I,1)
            CONTINUE
            THESE SINGULAR VALUES ARE THE LIMITS OF THE DISCRETE TIME EXTENDED
            INFINITE DIMENSIONAL 4ANKEL MATRIX
            PRINT,"FOR HELP IN ESTIMATION ON LIMITS OF ORDER"
            PRINT,"REDUCTION POSSIBILITY, THIS SINGULAR VALUE"
            PRINT,"FACTOR IS PRINTED OUT. TO UTILIZE, GROUP ALL"
            PRINT,"LARGE SINGULAR VALUES. THIS NUMBER IS"
            PRINT,"GENERALLY THE SMALLEST ONE CAN REDUCE THE SYSTEM"
            PRINT,"WITHOUT SIGNIFICANT LOSS OF ACCURACY."
            CALL PRINTM(M,1,MKAREF)

```

```

003340
003350
003360
003370
003380
003390
003400
003410
003420
003430
003440
003450
003460
003470
003480
003490
003500
003510
003520
003530
003540
003550
003560
003570
003580
003590
003600
003610
003620
003630
003640
003650
003660
003670
003680
003690

```

```

CALL LINVIF(SIG02,N,1,SIG02IN,1,WKAREA,IER)
003708
CALL MHULT(N,O,SIG02IN,WKAREA,N,N,N)
003710
CALL MHULT(WKAREA,WH,FINED,N,N,N)
003720
CALL MHULT(WINVED,SIG42,WKAREA,N,N,N)
003730
CALL MHULT(CBAR,WKAREA,CREAL,MC,NC,NC)
003740
CALL MHULT(A,WKAREA,WINVED,N,N,N)
003750
CALL LINVIF(WKAREA,H,1,TINV,1,WOFAREA,IER)
003760
CALL MHULT(TINV,WINVED,AREAL,MC,NC,NC)
003770
CALL MHULT(CINV,BBAR,BREAL,N,PD,ND)
003780
PRINT," "
003790
PRINT,"THE A' MATRIX IS"
003800
CALL PRINTM(N,H,AREAL)
003810
PRINT," "
003820
PRINT,"THE B' MATRIX IS"
003830
CALL PRINTM(MB,NB,BREAL)
003840
PRINT," "
003850
PRINT,"THE C' MATRIX IS"
003860
CALL PRINTM(MC,NC,CREAL)
003870
MF=MF+1
003880
PRINT," "
003890
CALL MHULT(CREAL,VINVTIAH,BBAR,MB,NB,NB)
003900
PRINT,"FOR OUTPUT THE 94T IS "
003910
CALL PRINTM(MB,NB,BBAR)
003920
CALL MHULT(VOUT,CREAL,CBAR,MC,NC,NC)
003930
PRINT,"FOR OUTPUT THE 94T IS "
003940
CALL PRINTM(MC,NC,CBAR)
003950
IF(IAM.EQ.1)GO TO 993
003960
DO 911=1,M
003970
DO 912=1,N
003980
WKAREA(I,J)=A(I,J)
003990
CONTINUE
004000
CALL LINVIF(WKAREA,N,1,WKAREA,1,WINVED,IER)
004010
CALL MHULT(C,WKAREA,FINED,MC,NC,NC)
004020
CALL MHULT(WINVED,B,WKAREA,MC,NC,NB)
004030
PRINT,"ENTER DESIRED ORDER OF REDUCED SYSTEM > "
004040
READ,10,DEF
004050
IF(DEF.EQ.1)GO TO 993
004060
IF(DEF.EQ.2)GO TO 993
004070
DO 921=1,10'DER
004080
DO 922=1,10'DER
004090

```

51

```

92      WINMED(I,J)=AKREAL(I,J)
        CONTINUE
        CALL LHMVIF(WINMED,IORDER,IU,NKAREA,1,TINV,IER)
        CALL LHMULT(CBAR,NORAREA,WINMED,MC,IORDER,IORDER)
        CALL LHMULT(WINMED,SBAR,NJFAKCA,MC,IORDER,NB)
        DO 93I=1,MC
          DO 93J=1,NB
            WINMED(I,J)=-(NKAREA(I,J)-NORAREA(I,J))
          CONTINUE
          PRINT*, "THE O' MATRIX IF BALANCED SYSTEM IS "
          PRINT*, "REDUCED TO ",IORDER," IS ."
          CALL F11TH(MC,NB,WINMED)
        CONTINUE
      9999  CONTINUE
          END
C-----
C      THIS SUBROUTINE READS IN A GIVEN MATRIX
C-----
C      SUBROUTINE READM(M,N,RRMAT)
      DIMENSION RRMAT(10,10)
      PRINT*, "PLEASE ENTER THE MATRIX BY ROWS"
      DO 100I=1,N,1
        WRITE(6,10) I
      10  FORMAT(14,"ROW(",I2,")=")
      20  READ*, (RRMAT(I,J),J=1,N,1)
      105  CONTINUE
      110  RETURN
      END
C-----
C      THIS SUBROUTINE ALLOWS USER TO CHANGE ERRONEOUS DATA
C-----
C      SUBROUTINE MISTAKE(M,N,RRMAT)
      DIMENSION RRMAT(10,10)
      PRINT*, "DO YOU WISH TO CHANGE ANY ELEMENTS?(1=YES,2=NO) > "
      MIST=1
      IF (JMOD(50,2)GO TO 2
      PRINT*, "TO CHANGE THE MATRIX ELEMENTS, YOU MAY TYPE"
      PRINT*, "THE ROW AND COLUMN SUBSCRIPTS AND THE CORRECTED"
      PRINT*, "MATRIX ELEMENT SEPARATED BY COMMAS > "
      MIST=1

```

```

004500
004510
004520
004530
004540
004550
004560
004570
004580
004590
004600
004610
004620
004630
004640
004650
004660
004670
004680
004690
004700
004710
004720
004730
004740
004750
004760
004770
004780
004790
004800
004810
004820
004830
004840
004850
004860
004870
004880
004890
004900

READ,MM,NN,CORREC
RMAI(MM,NN)=CORREC
PRINT,"ANY MORE? (YES,2=NO)
READ,INOU
IF(INOU.EQ.2)GO TO 2
GO TO 3
2 IF(MIST.EQ.1)CALL PRINTM(,N,RMAI)
RETURN
END
C THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
C
C SUBROUTINE PRINTM(M,N,DMAI)
C DIMENSION DMAI(10,10)
C DO 2 J=1,M
C WRITE (3,10) 1,(DMAI(I,J),J=1,N,1)
C FORMAT(140,12,(14,4(1P13.4,1X)))
C CONTINUE
C RETURN
C END
C THIS SUBROUTINE MULTIPLIES TWO MATRICES AND STORES IN C
C
C SUBROUTINE PMULT(A,B,L,M)
C DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
C DO 2 I=1,L
C DO 2K=1,M
C DO 1 J=1,M
C S1=SUM(A(I,J)*B(J,K)
C D(I,K)=SUM
C DO 3 I=1,L
C DO 3 J=1,M
C C(I,J)=D(I,J)
C RETURN
C END
C OVERLAY(1,0)
C THIS PROGRAM DISCRETIZES THE (A,B,C) SYSTEM AND PROVIDES THE

```



```

150 PRINT, "THE DISCRETE TIME CONTROLLABILITY"
    PRINT, "MATRIX IS "
    CALL PRINTN(N1,KNOD)
    THIS PART OF THE PROGRAM FINDS THE DISCRETE TIME
    C OBSERVABILITY MATRIX
    C
200 CONTINUE
    IF(IPASS.EQ.1)CALL O3(MC,NC,N,F,WINMED,WORAREA,RHOB,C,N2)
    IF(IPASS.EQ.2)CALL O3(MC,NC,N,F,WINMED,WORAREA,RHOB,CBAR,N2)
    IF(SUPP)GO TO 310
    PRINT, "THE DISCRETE TIME OBSERVABILITY "
    PRINT, "MATRIX IS "
    CALL PRINTN(N2,NR,N03)
    C THIS PART OF THE PROGRAM FINDS THE
    C DISCRETE HANKEL MATRIX
    C
300 CONTINUE
    CALL VMULFF(KMOR,KM07,N2,N,N1,30,10,HNKL,30,1EK)
    IF(SUPP)GO TO 410
    PRINT, "THE DISCRETE TIME HANKEL MATRIX IS "
    CALL PRINTN(N2,N1,HNKL)
    C
400 CONTINUE
    END
    *****
    C THIS SUBROUTINE MULTIPLIES TWO MATICES AND STORES IN C
    C *****
    SUBROUTINE MMULT(A,D,2,L,M,N)
    DIMENSION A(10,10),D(10,10),C(10,10),D(10,10)
    DO 21=1,L
    DO 22=1,M
    SUM=0.0
    DO 1J=1,N
    SUM=SUM+A(I,J)*B(J,K)
    D(I,K)=SUM
    DO 3I=1,L
    DO 3J=1,M
    C(I,J)=D(I,J)
    RETURN
    END
    *****

```

```

C----- THIS SUBROUTINE PRINTS OUT THE CONT AND OBS MATRICES
C-----
SUBROUTINE PRINTM1(M,4,E4AT)
  DIMENSION E4AT(30,10)
  DO 2 I=1,M,2
    WRITE (5,10) 1,(E4AT(I,J),J=1,M,1)
  10  FORMAT(140,12,(T,4(12E13.4,1X)))
  20  CONTINUE
  RETURN
  EN)

C----- THIS SUBROUTINE FINDS EXP(AHAT*TI) AND ITS INTEGRAL
C----- VIA A TRUNCATED INFINITE SERIES APPROXIMATION
C-----
SUBROUTINE FISKRET(N,1,DEL,EA,E4INT,NT)
  DIMENSION EA(10,10),E4INT(10,10),A(13,10),Z(10,10),
    CMINMED(1,1),CBAR(13,13)
  DO 13 I=1,N
    DO 13 J=1,N
      E4INT(I,J)=0.0
      F4INT(I,I)=DEL
      Z(I,J)=E4INT(I,J)
    CONTINUE
    DO 13 K=2,NT
      DO 13 I=1,N
        DO 13 J=1,N
          CBAR(I,J)=A(I,J)*(DEL/LOAT(K))
        CONTINUE
        CALL MMUL(7,CBAR,N,N,4,WINMED)
        DO 16 I=1,N
          DO 16 J=1,N
            Z(I,J)=WINMED(I,J)
          E4INT(I,J)=F4INT(I,J)+WINMED(I,J)
        CONTINUE
        CALL MMUL(4,E4INT,N,N,2)
        DO 50 I=1,N
          DO 50 J=1,N
            E4(I,J)=1.0
          E4(I,1)=1.0
        50  CONTINUE
      16  CONTINUE
    13  CONTINUE
  100  CONTINUE
  110  CONTINUE
  120  CONTINUE
  400  CONTINUE

```

```

005700
005710
005720
005730
005740
005750
005760
005770
005780
005790
005800
005810
005820
005830
005840
005850
005860
005870
005880
005890
005900
005910
005920
005930
005940
005950
005960
005970
005980
005990
006000
006010
006020
006030
006040
006050
006060
006070
006080
006090

```

```

110 CONTINUE
DO 100 I=1,N
DO 100 J=1,N
EA(I,J)=FA(I,J)+Z(I,J)
100 CONTINUE
RETURN
END

C-----
C THIS SUBROUTINE ACCEPTS EXP(AHAT) AND ITS INTEGRAL
C AND FINDS THE DISCRETE F MATRIX AND G MATRIX
C-----
SUBROUTINE FKNEET(F,G,N,N3,N4,NKAREA,A,B,I)
DIMENSION F(10,10),G(10,10),NKAREA(10,10),A(10,10)
DIMENSION B(10,10)
CALL DISCRET(N,A,I,F,AKAREA,50)
CALL FKNEET(NKAREA,B,G,N,N4,N7)
RETURN
END

C-----
C THIS SUBROUTINE FINDS THE DISCRETE CONTROLLABILITY
C MATRIX
C-----
SUBROUTINE COMBTY(F,G,M3,N8,N,N3,N4,NKAREA,WINNED,N1)
DIMENSION F(10,20),G(10,10),RM30(10,30),WINNED(10,10)
DIMENSION NKAREA(10,10)
DO 100 I=1,NP
DO 100 J=1,NP
RM30(I,J)=G(I,J)
CONTINUE
N1=N3
IF(N,N3,1)GO TO 150
CALL COMBTY(F,G,WINNED,N,N,N8)
DO 100 I=1,NP
DO 100 J=1,NP
K=J
N1=N1
RM30(I,N1+J)=WINNED(I,J)
CONTINUE
N1=N1+K
IF(N,N3,2)GO TO 150

```

```

006100
006110
006120
006130
006140
006150
006160
006170
006180
006190
006200
006210
006220
006230
006240
006250
006260
006270
006280
006290
006300
006310
006320
006330
006340
006350
006360
006370
006380
006390
006400
006410
006420
006430
006440
006450
006460
006470
006480
006490

```

```

006500
006510
006520
006530
006540
006550
006560
006570
006580
006590
006600
006610
006620
006630
006640
006650
006660
006670
006680
006690
006700
006710
006720
006730
006740
006750
006760
006770
006780
006790
006800
006810
006820
006830
006840
006850
006860
006870
006880
006890

N=N-1
DO 14, L=2, NN, 1
CALL MULT(F, WINMED, WDRAREA, N, N, N8)
DO 13, I=1, NP, 1
DO 13, J=1, NP, 1
NMJ(C, N1+J)=WDRAREA(I, J)
WINMED(I, J)=WDRAREA(I, J)
K=J
CONTINUE
N1=N1+K
CONTINUE
13J RETURN
END

C-----
C THIS SUBROUTINE FINDS THE DISCRETE OBSERVABILITY MATRIX
C-----
- SUBROUTINE OBS(MC, NC, N, F, WINMED, WDRAREA, KMOD, C, N2)
- DIMENSION WINMED(10,1), WDRAREA(10,1), C(10,10)
- DIMENSION F(10,10), KMOD(10,10)
DO 21, I=1, MC
DO 21, J=1, NC
KMOD(I, J)=C(I, J)
N2=MC
CONTINUE
IF(N.EQ.1) GO TO 250
CALL MULT(F, F, WINMED, MC, NC, N)
DO 22, I=1, MC
DO 22, J=1, NC
K=I
N2=MC
RMRB(N2+I, J)=WINMED(I, J)
CONTINUE
N2=N2+K
IF(N.EQ.0.2) GO TO 250
N1=N-1
DO 2, L=2, NP, 1
CALL MULT(WINMED, F, WDRAREA, MC, NC, N)
DO 23, I=1, MC
DO 23, J=1, NC
KMOD(I2+1, J)=WDRAREA(I, J)

```

```

      DIMHED(I,J)=HOKAREA(I,J)
      K=I
      220 CONTINUE
      N2=N2+K
      240 CONTINUE
      250 RETURN
      END
C-----
C THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
C-----
      SUBROUTINE PRANTH(M,N,DHAT)
      DIMENSION DHAT(10,10)
      DO 2 J=1,M,1
      WRITE(5,10) J, (DHAT(I,J),J=1,N,1)
      FORMAT(1H0,12,(14,4(1P E13.4,1X)))
      10 CONTINUE
      RETURN
      END
C-----
C THIS SUBROUTINE PRINTS OUT THE HANKEL MATRIX
C-----
      SUBROUTINE FRHIM2(M,N,DHAT)
      DIMENSION DHAT(20,30)
      DO 2 J=1,M,1
      WRITE(5,10) J, (DHAT(I,J),J=1,N,1)
      FORMAT(1H0,12,(14,4(1P E13.4,1X)))
      10 CONTINUE
      RETURN
      END
C-----
      OVERLAY(2,0)
C-----
C THIS PROGRAM PLOTS AS WELL AS LISTS THE DISCRETE-TIME
C CONTROLLABILITY, OBSERVABILITY AND HANKEL MATRICES SINGULAR
C VALUES AS A FUNCTION OF SAMPLING TIME. THIS IS USEFUL TO THE
C USER AS ANOTHER CONSTRAINT ON THE CHOICE OF APPROPRIATE SAMPLE
C TIME FOR THE SYSTEM IN CONSIDERATION
C-----
      PROGRAM PLOTSIV

```

```

006900
006910
006920
006930
006940
006950
006960
006970
006980
006990
007000
007010
007020
007030
007040
007050
007060
007070
007080
007090
007100
007110
007120
007130
007140
007150
007160
007170
007180
007190
007200
007210
007220
007230
007240
007250
007260
007270
007280
007290

```


21

AD-A080 371

AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 20/4
MODEL ORDER REDUCTION USING THE BALANCED STATE REPRESENTATION: --ETC(U)
DEC 79 J R MCCLENDON

UNCLASSIFIED

AFIT/GE/EE/79-22

NL

3 OF 3

AD-A080 371

END

DATE

FILED

3-80

REF

```

260 CALL TTLES(I09(13))
    PRINT," "
    PRINT,"NOTE: THE CALCULATION DELTA FOR THIS ALGORITHM"
    PRINT,"IS THE INITIAL TIME. THEREFORE THE INITIAL TIME MUST"
    PRINT,"BE LESS THAN ONE. THE FINAL TIME MUST BE WITHIN"
    PRINT,"10 STEPS OF THE INITIAL TIME"
    PRINT,"PLEASE ENTER THE BEGINNING SAMPLING TIME"
    PRINT,"AND THE FINAL SAMPLING TIME > "
    READ,TINIT,TF
    TDEL=TIME
    IF(SUP)PRINT,"EVEN WITH LISTING SUPPRESSED THIS OPTION"
    IF(SUP)PRINT,"TAKES TIME--DEPENDENT ON NUMBER OF POINTS"
    NT=(TF-TINIT)/TDEL
    T=INIT
    TMAX(1)=T
    CALL DRESET(F,G,N,MB,40,MKAKEA,A,B,TMAX(1))
    CALL CORRTY(F,G,MB,N7,N,R400,MOKAPEA,MIMELD,N1)
    CALL DRS(40,NC,N,F,WINMED,MOKRCA,MKOB,C,N2)
    DO 211 I=1,N
    211 CONTINUE
    RMCC(JJ,III)=RMCC(III,JJ)
    CONTINUE
    CALL LSVALR(RMCC,N1,N,3,30,7,MINMED,SIGC,UU,VV)
    CALL LSVALP(RMCC,N2,N,3,30,6,MINMED,SIGO,UU,VV)
    IF(ILIKE.EQ.3.AND.ILIKE.NE.4)GO TO 206
    CALL OVERLAY(4,MIMO,4,1)
    CALL VSO(ITA(SIGC,N)
    CALL VSO(ITA(SIGO,N)
    SC1AX(1)=SIGC(JWAS,1)
    SC1IN(1)=SIGC(IWAS,1)
    SC1AX(1)=SIGC(JWAS,1)
    SC1IN(1)=SIGC(IWAS,1)
    COMD(1)=SC1IN(1)/SCMAX(1)
    COMD(1)=SC1IN(1)/SUMAX(1)
    IF(SUP)GO TO 100
    IF(ILIKE.EQ.2.AND.ILIKE.NE.3)GO TO 300
    PRINT,"THE SINGULAR VALUES OF THE CONTROLLABILITY"
    PRINT,"MATRIX FOR TIME > ",TMAX(1)," ARE "
    CALL PRINTN(N,1,SIGC)
    PRINT," "

```

```

000900
008510
006520
006530
006540
006550
006560
006570
006580
006590
006600
006610
006620
006630
006640
006650
006660
006670
006680
006690
006700
006710
006720
006730
006740
006750
006760
006770
006780
006790
006800
006810
006820
006830
006840
006850
006860
006870
006880
006890

```

```

300  IF(ILIKE.EQ.1.OF.ILIKE.EQ.3)GO TO 100
      PRINT,"THE SINGULAR VALUES OF THE OBSERVABILITY"
      PRINT,"MATRIX FOR TIME > ",T*1(1)," ARE "
      CALL PRINTM(N,1,SIG)
100  CONTINUE
      I=T+100
      DO 20 J=1,N
      DO 25 J=1,N
      JC(J,J)=F(I,J)
      UCTRAN(I,J)=WKAREA(I,J)
      VOUT(I,J)=F(I,J)
      VUJ(I,J)=F(I,J)+1.
      CONTINUE
      T*1(2)=T
      CALL MATIA(UCTRAN,VOUT,N,N,N,N,VOUT)
      CALL MULTI(F,UC,WKAREA,N,N,N)
      DO 30 J=1,N
      DO 7 N=1,N
      F(LL,N)=WOFAREA(LL,N)
      CONTINUE
      CALL MAT1(VOUT,R,N,N,N,N,J)
      CALL COMITY(F,G,PD,H7,N,T*DD,WCPAREA,WINMED,N1)
      CALL JDS(MC,NC,N,F,WINMED,WKAREA,RHOB,C,N2)
      DO 31 I=1,N
      DO 3 JJ=1,N1
      RMCC(JJ,I)=RHOB(I,I,J)
      CONTINUE
      CALL LVALP(RMCC,N1,N,3,3,0,WKAREA,SIGC,UU,VV)
      CALL LVALP(RHOB,N2,N,3,3,0,WKAREA,SIGO,UU,VV)
      IF(ILIKE.NE.3.AND.ATC.NE.4)GO TO 200
      CALL ONECLAY(4,H*MO,4,2)
      CALL VDATA(SIGC,N)
      CALL VDATA(SIGO,N)
      SC4X(2)=SIGC(JWAS,1)
      SC4X(2)=SIGO(JWAS,1)
      SC41(12)=SIGC(IWAS,1)
      SC41(12)=SIGO(IWAS,1)
      SC42(12)=SIGC(IWAS,1)
      SC42(12)=SIGO(IWAS,1)
      SC43(12)=SIGC(IWAS,1)
      SC43(12)=SIGO(IWAS,1)
      IF(SUP)GO TO 200

```

D-24


```

10  FORMAT (1H0,12, (T4,4(12F13.4,1X)))
20  CONTINUE
    RETURN
    END
C-----
C   THIS SUBROUTINE MULTIPLIES TWO MATRICES AND STORES IN C
C-----
SUBROUTINE MMULT(A,B,2,L,M,N)
  DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
  DO 2 I=1,L
    DO 2 K=1,M
      SUM=0.000
      DO 1 J=1,N
        SUM=SUM+A(I,J)*B(J,K)
      1  C(I,K)=SUM
      DO 3 I=1,L
        DO 3 J=1,N
          C(I,J)=D(I,J)
        3  RETURN
      END
C-----
C   THIS SUBROUTINE PRINTS OUT THE CONT AND URS MATRICES
C-----
SUBROUTINE PRINTM(N,N,E4BT)
  DIMENSION EMAT(30,10)
  DO 2 I=1,N*2
    WRITE (5,10) I,(EMAT(I,J),J=1,N,1)
  10  FORMAT (1H0,12, (T4,4(12F13.4,1X)))
  20  CONTINUE
    RETURN
    END
C-----
C   THIS SUBROUTINE FINDS EXP(AMAT*T) AND ITS INTEGRAL
C   VIA A TRUNCATED INFINITE SERIES APPROXIMATION
C-----
SUBROUTINE DISKBEI(N,A,DEL,CA,EAINI,NT)
  DIMENSION EA(10,10),EINT(10,10),A(10,10),Z(10,10),
  C(I,J)=1,N
  DO 1 J=1,N

```

```

010100
010110
010120
010130
010140
010150
010160
010170
010180
010190
010200
010210
010220
010230
010240
010250
010260
010270
010280
010290
010300
010310
010320
010330
010340
010350
010360
010370
010380
010390
010400
010410
010420
010430
010440
010450
010460
010470
010480
010490

```

```

100      EAINT(I,J)=.000
        EAINT(I,I)=FEL
        Z(I,J)=EAINT(I,J)
        CONTINUE
        DO 130 K=2,NZ
        DO 131 I=1,N
        DO 132 J=1,M
        C3AP(I,J)=A(I,J)*(DEL/FLORT(K))
        CONTINUE
        CALL MMULT(7,CBAP,WIMFEO,M,N,M)
        DO 133 I=1,M
        DO 134 J=1,N
        Z(I,J)=WIMFEO(I,J)
        EAINT(I,J)=EAINT(I,J)+WIMFEO(I,J)
        CONTINUE
        CALL MMULT(8,EAINT,Z,N,N,N)
        DO 135 I=1,N
        DO 136 J=1,M
        EA(I,J)=Z(I,J)
        EA(I,J)=1.0/6
        CONTINUE
        DO 137 I=1,N
        DO 138 J=1,M
        EA(I,J)=EA(I,J)+Z(I,J)
        CONTINUE
        RETURN
        EN
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```

D-28

```

00 20 I=1,MC
00 21 J=1,NC
      K=J*(I-1)+1
      N2=N2+K
      CONTINUE
      IF(N.EQ.1) GO TO 250
      CALL MULT(C,F,WINMED,PC,NC,N)
      DO 22 I=1,MC
      DO 23 J=1,NC
        K=I
        N2=MC
        RMJW(N2+I,J)=WINMED(I,J)
        CONTINUE
        IF(N.EQ.2) GO TO 250
        IN=N-1
        DO 24 L=1,MN,1
          CALL MULT(WINMED,F,WKAREA,MC,NC,N)
          DO 25 I=1,MC
          DO 26 J=1,NC
            KMJW(N2+I,J)=WKAREA(I,J)
            WINMED(I,J)=WKAREA(I,J)
            K=I
          CONTINUE
          N2=N2+K
        CONTINUE
        RETURN
      EN
C-----
C THIS SUBROUTINE PLOTS A HORIZONTAL GRAPH
C-----
      SUBROUTINE HGRAPH(X,Y,N,IO,NO,NP,NS)
      DIMENSION X(1),Y(1),IO(1)
      IF(NJ.LT.1) GO TO 10
      CALL SCALE(Y,7,MN,1)
      CALL PLOT(M.5,J,-3)
      CALL PLOT(0.,11.93)
      CALL PLOT(-1.35,1.35,3)
      CALL PLOT(-.15,1.35,2)
      IF(IO(1).EQ.0) GO TO 20
      CALL PLOT(-.05,2.55,3)
      CALL PLOT(-7.05,7.55,2)
      CALL PLOT(-7.05,7.55,2)

```

```

011390
011310
011320
011330
011340
011350
011360
011370
011380
011390
011400
011410
011420
011430
011440
011450
011460
011470
011480
011490
011500
011510
011520
011530
011540
011550
011560
011570
011580
011590
011600
011610
011620
011630
011640
011650
011660
011670
011680
011690

```

```

20 DO 2J=1,7,2
   CALL SYM3OL(I*-1.05,7.55,07,10(I),90.,20)
   CALL PLOT(-1.05,7.55,3) 3CALL PLOT(-0.05,7.55,2)
   CALL PLOT(-0.05,9.55,2) 3 CALL PLOT(-7.05,9.55,2)
   CALL PLOT(-7.15,9.55,3)
25 CALL PLOT(-1.35,9.55,2) 3 CALL PLOT(-1.35,1.35,2)
   CALL SYM3OL(-6.55,1.35,1,10(I),0.,2)
   CALL AXIS(-1.05,2.1,3),-20,7.50,X(N+1),X(N+2))
   CALL AXIS(-1.05,2.1,3),2.5,18.5,Y(N+1),Y(N+2))
   Y(1+2)=-Y(N+2)
30 X(1+1)=X(N+2)-2.1-X(N+2) 1 Y(N+1)=Y(N+1)+1.05-Y(N+2)
   CALL LINE(Y,X,N,1,MP,NS)
   X(1+1)=X(N+1)+2.1-X(N+2) 3 Y(N+1)=Y(N+1)-1.05-Y(N+2)
   Y(1+2)=-Y(N+2)
   RETURN 3 END
C-----
C THIS SUBROUTINE PLOTS A VERTICAL GRAPH
C-----
SUBROUTINE VGRAPH(X,Y,N,ID,HO,MP,NS)
DIMENSION X(1),Y(1),ID(1) 3 I=(HO.EQ.2)GO TO 30
IF(HO.EQ.0)GO TO 10
CALL SCALE(Y,4,2,H,1) 3 CALL SCALE(Y,7,N,1)
CALL PLOT(8,5,0,-3) 3 CALL PLOT(0,11,3)
CALL PLOT(-1.35,1.35,3)
CALL PLOT(-7.15,1.35,2) 3 CALL PLOT(-7.15,9.55,2)
CALL PLOT(-1.35,9.55,2) 3 IF(10(I).EQ.0)GO TO 25
CALL PLOT(-1.05,9.55,3) 3 CALL PLOT(-3.45,9.55,2)
DO 2J=1,7,2
CALL SYM3OL(-3.15,9.55,1,10(I),0.,10)
CALL PLOT(-3.45,9.55,2) 3 CALL PLOT(-3.45,9.55,2)
CALL PLOT(-1.05,9.55,2) 3 CALL PLOT(-1.05,9.55,2)
CALL PLOT(-1.35,9.55,3)
CALL SYM3OL(-6.55,1.35,1,10(I),0.,3)
CALL AXIS(-7.15,1.35,1,3),-20,7.50,X(N+1),X(N+2))
CALL AXIS(-7.15,1.35,1,3),2.5,18.5,Y(N+1),Y(N+2))
X(1+1)=X(N+1)+0.7-X(N+2) 3 Y(N+1)=Y(N+1)-1.05-Y(N+2)
CALL LINE(X,Y,N,1,MP,NS)
X(1+1)=X(N+1)-0.7-X(N+2) 1 Y(N+1)=Y(N+1)+1.05-Y(N+2)
RETURN 3 END

```

```

011700
011710
011720
011730
011740
011750
011760
011770
011780
011790
011800
011810
011820
011830
011840
011850
011860
011870
011880
011890
011900
011910
011920
011930
011940
011950
011960
011970
011980
011990
012000
012010
012020
012030
012040
012050
012060
012070
012080
012090

```



```

CO=HOF/MIHOF/A(10,10),M,N,B(10,10),MO,NB,F(10,10),
*G(10,10),C(10,10),MC,NC
CU=HOF/PLT(1/1,SIVS(5C,1),SUP,IWAS,JWAS1
CALL VMULFF(KMOF,HMOD,N2,N,1,30,10,HMKL,3C,IER)
IF(N2-GE,N1)CALL LSVA-R(HMKL,N2,N1,30,C,VV,SIVS,UU,VV)
DO 5 JI=1,N2
DO 5 JJ=1,N1
HMKL(JJ,II)=HMKL(II,JJ)
CU=INDE
IF(N2-LE,N1)CALL LSVA-R(HMKL,N1,N2,30,0,VV,SIVS,UU,VV)
IF(N2-GE,N1)CALL VSORFA(SIVS,N1)
IF(N2-LE,N1)CALL VSORFA(SIVS,N2)
IF(N2-GE,N1)NUMEN1
IF(N2-LE,N1)NUMEN2
IWAS1=TW17
JWAS1=JWAS
JWAS1=JWAS
IF(SUP)GO TO 4JL0
PRINT, 'THE SINGULAR VALUES OF THE HANKEL MAT. FOR TIME ',
+TAT(1), ' ARE >'
IF(N2-GE,N1)CALL PRINTM2(N1,1,SIVS)
IF(N2-LE,N1)CALL PRINTM2(N2,1,SIVS)
4JL0 CONTINUE
IF(N2-GE,N1)HMAX(1)=SIVS(JWAS1,1)
IF(N2-LE,N1)HMAX(1)=SIVS(JWAS1,1)
HMIN(1)=SIVS(IWAS1,1)
HMIN(1)=HMIN(1)/HMAX(1)
END
C-----
C THIS SUBROUTINE PRINTS OUT THE HANKEL MATRIX SINGULAR VALUES
C-----
SUBROUTINE PRINTM2(M,N,OMAT)
DIMENSION D(10,1)
DO 20 J=1,M,1
WRITE(6,1) I, (OMAT(I,J),J=1,N,1)
FORMAT(140,12,(14,4(1P13.4,1X)))
20 CONTINUE
RETURN
END
C-----
OVERLAY(4,2)

```

012500
012510
012520
012530
012540
012550
012560
012570
012580
012590
012600
012610
012620
012630
012640
012650
012660
012670
012680
012690


```

DO 207 I=1,M,1
  WRITE(6,10) I,(OMAT(I,J),J=1,N,1)
  FORMAT(140,12,(14,4,(12E13.4,1X)))
20 CONTINUE
  RETURN
ENJ
C *****
OVERLAY(1,C)
C *****
C THIS PROGRAM FINDS THE CONTINUOUS TIME CONTROLLABILITY
C AND OBSERVABILITY MATRICES.
C
C
PROGRAM CONTIME
COMMON/MAINI/NOIH,NOI11,COM1/INOI/KIN,KOUT,KPUNCH
COMMON/12HOF/RHOB(30,10),RHOD(10,30),RMCO(30,10),N1,N2,UU(30,30)
C,VV(1,30),WKL(30,30),H,KLF(32,32)
COMMON/MIMO/MOAKREA(10,10),MKAREA(17,10),WINMED(10,10)
COMMON/MIMO/A(10,10),B(10,10),MB,NB,F(10,10),G(10,10),C(10,10),16,13,5,0
),10,10
CALL COPYTY(A,B,MB,NB,N,CMDD,MOAKREA,WINMED,N1)
CALL SRC(MC,NC,N,A,WINKL),H,KAREA,KMOB,C,N2)
PRINT,"THE CONTINUOUS TIME CONTROLLABILITY MATRIX IS"
CALL PRINTMN,M,RHOD
PRINT,"THE CONTINUOUS TIME OBSERVABILITY MATRIX IS"
CALL PRINTMN(N2,N,RHOB)
END
C *****
C THIS SUBROUTINE PRINTS OUT MATRICES
C *****
SUBROUTINE PRINTM(M,N,DAT)
  DIMENSION DMAT(10,10)
  DO 201 I=1,M,1
    WRITE(6,10) I,(DMAT(I,J),J=1,N,1)
    FORMAT(140,12,(14,4,(12E13.4,1X)))
20 CONTINUE
  RETURN
END
C *****
C THIS SUBROUTINE PRINTS OUT THE OBSERVABILITY MATRIX
C *****

```

```

C..... SUBROUTINE PRINTM1(M,4,E4AT)
DIMENSION F(4,10)
DO 277 I=1,M,1
WRITE(5,10) I,(EMAT(I,J),J=1,N,1)
FORMAT(14G,72,(T4,4(12F13.4,1X)))
27 CONTINUE
RETURN
END
C..... THIS SUBROUTINE FINDS THE CONTINUOUS TIME CONTROLLABILITY MATRIX
C.....
SUBROUTINE CONSTY(F,G,M3,NB,M3HDD,M3FAREA,WINMED,M1)
DIMENSION F(10,10),G(10,10),M3D(10,30),WINMED(10,10)
DIMENSION M3FAREA(10,10)
DO 11 I=1,M3
DO 12 J=1,NB
M3D(I,J)=G(I,J)
11 CONTINUE
M1=M3
12 DO 13 I=1,M3 TO 100
CALL MPULT(F,G,WINMED,M,I,NB)
DO 12 I=1,M3
DO 12 J=1,NB
M1=M1+M3D(I,J)
13 CONTINUE
123 CALL MPULT(F,WINMED,M3FAREA,N,N,NB)
DO 13 I=1,M3,1
DO 13 J=1,NB,1
M3D(I,J)=M3FAREA(I,J)
WINMED(I,J)=M3FAREA(I,J)
M1=M1+M3D(I,J)
13 CONTINUE
M1=M1+M
134

```

```

014100
014110
014120
014130
014140
014150
014160
014170
014180
014190
014200
014210
014220
014230
014240
014250
014260
014270
014280
014290
014300
014310
014320
014330
014340
014350
014360
014370
014380
014390
014400
014410
014420
014430
014440
014450
014460
014470
014480
014490

```

```

140 CONTINUE
150 RETURN
160 END
C-----
C THIS SUBROUTINE FINDS THE CONTINUOUS OBSERVABILITY MATRIX
C-----
SUBROUTINE (9S(MC,NC,4,F,WINMED,HORAREA,RHOB,C,N2)
DIMENSION WINMED(10,10),HORAREA(10,10),C(10,10)
DO 200 I=1,MC
DO 200 J=1,NC
200 J(J,I)=C(I,J)
N2=MC
CONTINUE
IF(NC.EQ.1)GO TO 250
CALL MULT(F,F,WINMED,MC,NC,N)
DO 220 I=1,MC
DO 220 J=1,NC
220 J(J,I)=F
N2=MC
RMJ0(N2*I,J)=WINMED(I,J)
CONTINUE
N2=N2*N
IF(NC.EQ.2)GO TO 250
N2=N2-1
DO 240 L=2,NC,1
CALL MULT(WINMED,F,HORAREA,MC,NC,N)
DO 230 I=1,MC
DO 230 J=1,NC
RMJ3(N2*I,J)=HORAREA(I,J)
WINMED(I,J)=HORAREA(I,J)
N2=I
CONTINUE
N2=N2*N
240 CONTINUE
250 RETURN
260 END
C-----
C THIS SUBROUTINE MULTIPLIES TWO MATRICES
C-----

```

```

014500
014510
014520
014530
014540
014550
014560
014570
014580
014590
014600
014610
014620
014630
014640
014650
014660
014670
014680
014690
014700
014710
014720
014730
014740
014750
014760
014770
014780
014790
014800
014810
014820
014830
014840
014850
014860
014870
014880
014890

```



```

C-----
C THIS SUBROUTINE READS IN A GIVEN MATRIX
C-----
SUBROUTINE READM(M,N,R,MAT)
  DIMENSION R(MAT(10,10))
  PRINT *, "PLEASE ENTER THE MATRIX BY ROWS"
  DO I=1,M
    READ *, R(I,1:N)
  END DO
  WRITE(6,15) I
  FORMAT(14F10.2, "ROW(", I2, ")=")
  READ *, R(MAT(10,10))
  CONTINUE
  RETURN
END

C-----
C THIS SUBROUTINE ALLOWS USER TO CHANGE ERRONEOUS DATA
C-----
SUBROUTINE MISTAKE(M,I,R,MAT)
  DIMENSION R(MAT(10,10))
  MIST=0
  PRINT *, "DO YOU WISH TO CHANGE ANY ELEMENTS? (1=YES, 2=NO) > "
  READ *, INDIC
  IF (INDIC.EQ.2) GO TO 2
  PRINT *, "TO CHANGE THE MATRIX ELEMENTS, YOU MAY TYPE"
  MIST=1
  PRINT *, "THE ROW AND COLUMN SUBSCRIPTS AND THE CORRECTED"
  PRINT *, "MATRIX ELEMENT SEPARATED BY COMMAS > "
  READ *, M,N,CORREC
  PRINT *, "ANY MORE (1=YES, 2=NO) > "
  READ *, INDU
  DO TO 3
  RETURN
END

C-----
C THIS SUBROUTINE PRINTS OUT A GIVEN MATRIX
C-----
SUBROUTINE PRINTM(M,N,MAT)
  DIMENSION MAT(10,10)

```

```

015700
015710
015720
015730
015740
015750
015760
015770
015780
015790
015800
015810
015820
015830
015840
015850
015860
015870
015880
015890
015900
015910
015920
015930
015940
015950
015960
015970
015980
015990
016000
016010
016020
016030
016040
016050
016060
016070
016080
016090

```



```

16      'READ',IMAY
      GO TO (1,2,3,4,5,6,7),NEX1
      CALL LSVALR(A,N,N,1,10,1,MKAREA,WINMED,UH,VHTRAN)
      CALL PRINTS(IMAY,UH,V4TRAN,N,N,10)
      GO TO 50

20      CALL LSVALR(ANEAL,N,N,10,10,1,MKAREA,WINMED,UH,VHTRAN)
      CALL PRINTS(IMAY,UH,V4TRAN,N,N,10)
      GO TO 50

30      CALL LSVALR(F,N,N,1,10,1,MKAREA,WINMED,UH,VHTRAN)
      CALL PRINTS(IMAY,UH,V4TRAN,N,N,10)
      GO TO 50

40      GO 12,I=1,N2
      DO 13,J=1,N1
      MNKL(I,J,I)=MNKL(I,J)
      CONTINUE
      IF(N2.LE.N1)CALL LSVALR(F,MNKL,N1,N2,30,1,MNKL,SIVS,UU,VV)
      IF(N2.GT.N1)CALL LSVALR(MNKL,N2,N1,30,1,MNKL,SIVS,UU,VV)
      GO TO 50

      PRINT,"ENTER ROW DIMENSION OF INPUT MATRIX > "
      READ,NPP
      PRINT,"ENTER COLUMN DIMENSION OF INPUT MATRIX > "
      READ,NPP
      CALL SEATN(MPP,NPP,PP4L(I))
      CALL PRINTM(MPP,NPP,PP4AT)
      CALL M3STAKE(MPP,NPP,PP4AT)
      IF(NPP.EQ.1)GO TO 50
      DO 12,I=1,NPP
      DO 13,J=1,NPP
      PP4AT(I,J,I)=PP4AT(I,J)
      CONTINUE
      CALL LSVALR(PP4AT,N,N,F,MPP,10,10,1,MKAREA,WINMED,UH,VHTRAN)
      GO TO 50

42      CALL LSVALR(PP4AT),MPP,NPP,10,10,1,MKAREA,WINMED,UH,VHTRAN)
      PRINT,"THE NON-ZERO SINGULAR VALUES ARE:"
      IF(NEX1.EQ.1)AND(N2.LE.N1)CALL PRINTM1(N2,1,SIVS)
      IF(NEX1.EQ.1)AND(N2.GT.N1)CALL PRINTM1(N1,1,SIVS)
      IF(MPP.EQ.1)AND(NPP.LE.NPP)CALL PRINTM(MPP,1,WINMED)
      IF(MPP.GT.1)AND(NPP.GT.NPP)CALL PRINTM(NPP,1,WINMED)
      IF(MPP.EQ.1)AND(NEX1.EQ.1)CALL PRINTM(N,1,WINMED)
      IF(NEX1.EQ.1)AND(N2.LE.N1)CALL PRINTM1(N2,1,SIVS)

```

016500
016510
016520
016530
016540
016550
016560
016570
016580
016590
016600
016610
016620
016630
016640
016650
016660
016670
016680
016690
016700
016710
016720
016730
016740
016750
016760
016770
016780
016790
016800
016810
016820
016830
016840
016850
016860
016870
016880
016890

```

016900 IF(NEX1.EQ.1.AND.N2.GT.N1)CALL PRINTS1(IMAY,UU,VV,N2,N1,3J)
016910 IF(NEX1.EQ.1.AND.NPP.E.NPP)CALL PRINTS(IMAY,UM,VHTRAM
016920 +,PP,PP,1N)
016930 IF(NEX1.EQ.1.AND.MPP.E.MPP)CALL PRINTS(IMAY,UM,VHTRAM
016940 +,MPP,MPP,1N)
016950 END
016960 SUBROUTINE PRINTS1(IMAY,JJ,JV,N1,N2,1NKK)
016970 DIMENSION UU(1NKK,1),VV(1NKK,1)
016980 IF(IMAY.EQ.2.OU.IMAY.EQ.4)PRINT,"THE LEFT SINGULAR VECTORS"
016990 IF(1IMAY.EQ.2.OU.IMAY.EQ.4)CALL PRINTM1(N1,N2,UU)
017000 IF(1IMAY.EQ.2.OU.IMAY.EQ.4)PRINT,"THE RIGHT SINGULAR VECTORS"
017010 IF(1IMAY.EQ.2.OU.IMAY.EQ.4)CALL PRINTM1(N2,N2,VV)
017020 RETURN
017030 END
017040 C-----
017050 C THIS SUBROUTINE PRINTS OUT SINGULAR VECTORS
017060 C-----
017070 SUBROUTINE PRINTS(IMAY,UU,VV,N1,N2,1NKK)
017080 DIMENSION UU(1NKK,1),VV(1NKK,1)
017090 IF(1IMAY.EQ.2.OU.IMAY.EQ.4)PRINT,"THE LEFT SINGULAR"
017100 IF(1IMAY.EQ.2.OU.IMAY.EQ.4)PRINT,"VECTORS ARE "
017110 IF(1IMAY.EQ.2.OU.IMAY.EQ.4)CALL PRINTM1(N1,N2,UU)
017120 IF(1IMAY.EQ.2.OU.IMAY.EQ.4)PRINT,"THE RIGHT SINGULAR"
017130 IF(1IMAY.EQ.2.OU.IMAY.EQ.4)PRINT,"VECTORS ARE "
017140 IF(1IMAY.EQ.2.OU.IMAY.EQ.4)CALL PRINTM1(N2,N2,VV)
017150 RETURN
017160 END
017170 C-----
017180 C THIS SUBROUTINE READS IN A GIVEN MATRIX
017190 C-----
017200 SUBROUTINE READM(M,N,RRM1)
017210 DIMENSION RRM1(10,10)
017220 PRINT,"ENTER THE MATRIX BY ROWS"
017230 DO 115 I=1,M,1
017240 WRITE(I,10) I
017250 FORMAT(10,"ROW(",12,")=")
017260 READ(1,RRM1(I,J),J=1,N,1)
017270 CONTINUE
017280 RETURN
017290 END

```

```

C.....
C THIS SUBROUTINE ALLOWS USER TO CHANGE EXPONEOUS DATA
C.....
SUBROUTINE MISTAKE(M,R,IAT)
  DIMENSION R(10,10)
  MIST=2
  PRINT*, "DO YOU WISH TO CHANGE ANY ELEMENTS?(1=YES,2=NO) > "
  IF(1) GO TO 2
  PRINT*, "TO CHANGE THE MATRIX ELEMENTS, YOU MAY TYPE"
  MIST=1
  PRINT*, "THE ROW AND COLUMN SUBSCRIPTS AND THE CORRECTED"
  PRINT*, "MATRIX ELEMENT SEPARATED BY COMMAS > "
  READ*, M, N, CORREC
  R(1,1)=CORREC
  PRINT*, "ANY MORE?(1=YES,2=NO) > "
  IF(1) GO TO 2
  GO TO 3
  IF(MIST=2) CALL PRINTM(M,N,R,IAT)
  RETURN
END
C.....
C THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
C.....
SUBROUTINE PRINTM(M,N,DAT)
  DIMENSION DAT(10,10)
  DO I=1,M
    WRITE(0,10) I, (DAT(I,J), J=1,N,1)
  END DO
  FORMAT(1H,12,(T4,1PE15.4,1X))
  CONTINUE
  RETURN
END
C.....
C THIS SUBROUTINE PRINTS OUT THE HANKEL MATRIX SING VALUES
C.....
SUBROUTINE PRINTM1(M,N,DAT)
  DIMENSION DAT(30,1)
  DO I=1,M
    WRITE(0,10) I, (DAT(I,J), J=1,N,1)
  END DO
  FORMAT(1H,12,(T4,1PE15.4,1X))
  CONTINUE
  RETURN
END

```

```

017308
017310
017312
017314
017316
017318
017320
017322
017324
017326
017328
017330
017332
017334
017336
017338
017340
017342
017344
017346
017348
017350
017352
017354
017356
017358
017360
017362
017364
017366
017368
017370
017372
017374
017376
017378
017380
017382
017384
017386
017388
017390
017392
017394
017396
017398
017400
017402
017404
017406
017408
017410
017412
017414
017416
017418
017420
017422
017424
017426
017428
017430
017432
017434
017436
017438
017440
017442
017444
017446
017448
017450
017452
017454
017456
017458
017460
017462
017464
017466
017468
017470
017472
017474
017476
017478
017480
017482
017484
017486
017488
017490
017492
017494
017496
017498
017500
017502
017504
017506
017508
017510
017512
017514
017516
017518
017520
017522
017524
017526
017528
017530
017532
017534
017536
017538
017540
017542
017544
017546
017548
017550
017552
017554
017556
017558
017560
017562
017564
017566
017568
017570
017572
017574
017576
017578
017580
017582
017584
017586
017588
017590
017592
017594
017596
017598
017600
017602
017604
017606
017608
017610
017612
017614
017616
017618
017620
017622
017624
017626
017628
017630
017632
017634
017636
017638
017640
017642
017644
017646
017648
017650
017652
017654
017656
017658
017660
017662
017664
017666
017668
017670
017672
017674
017676
017678
017680
017682
017684
017686
017688
017690
017692
017694
017696
017698
017700
017702
017704
017706
017708
017710
017712
017714
017716
017718
017720
017722
017724
017726
017728
017730
017732
017734
017736
017738
017740
017742
017744
017746
017748
017750
017752
017754
017756
017758
017760
017762
017764
017766
017768
017770
017772
017774
017776
017778
017780
017782
017784
017786
017788
017790
017792
017794
017796
017798
017800
017802
017804
017806
017808
017810
017812
017814
017816
017818
017820
017822
017824
017826
017828
017830
017832
017834
017836
017838
017840
017842
017844
017846
017848
017850
017852
017854
017856
017858
017860
017862
017864
017866
017868
017870
017872
017874
017876
017878
017880
017882
017884
017886
017888
017890
017892
017894
017896
017898
017900
017902
017904
017906
017908
017910
017912
017914
017916
017918
017920
017922
017924
017926
017928
017930
017932
017934
017936
017938
017940
017942
017944
017946
017948
017950
017952
017954
017956
017958
017960
017962
017964
017966
017968
017970
017972
017974
017976
017978
017980
017982
017984
017986
017988
017990
017992
017994
017996
017998
018000
018002
018004
018006
018008
018010
018012
018014
018016
018018
018020
018022
018024
018026
018028
018030
018032
018034
018036
018038
018040
018042
018044
018046
018048
018050
018052
018054
018056
018058
018060
018062
018064
018066
018068
018070
018072
018074
018076
018078
018080
018082
018084
018086
018088
018090
018092
018094
018096
018098
018100
018102
018104
018106
018108
018110
018112
018114
018116
018118
018120
018122
018124
018126
018128
018130
018132
018134
018136
018138
018140
018142
018144
018146
018148
018150
018152
018154
018156
018158
018160
018162
018164
018166
018168
018170
018172
018174
018176
018178
018180
018182
018184
018186
018188
018190
018192
018194
018196
018198
018200
018202
018204
018206
018208
018210
018212
018214
018216
018218
018220
018222
018224
018226
018228
018230
018232
018234
018236
018238
018240
018242
018244
018246
018248
018250
018252
018254
018256
018258
018260
018262
018264
018266
018268
018270
018272
018274
018276
018278
018280
018282
018284
018286
018288
018290
018292
018294
018296
018298
018300
018302
018304
018306
018308
018310
018312
018314
018316
018318
018320
018322
018324
018326
018328
018330
018332
018334
018336
018338
018340
018342
018344
018346
018348
018350
018352
018354
018356
018358
018360
018362
018364
018366
018368
018370
018372
018374
018376
018378
018380
018382
018384
018386
018388
018390
018392
018394
018396
018398
018400
018402
018404
018406
018408
018410
018412
018414
018416
018418
018420
018422
018424
018426
018428
018430
018432
018434
018436
018438
018440
018442
018444
018446
018448
018450
018452
018454
018456
018458
018460
018462
018464
018466
018468
018470
018472
018474
018476
018478
018480
018482
018484
018486
018488
018490
018492
018494
018496
018498
018500
018502
018504
018506
018508
018510
018512
018514
018516
018518
018520
018522
018524
018526
018528
018530
018532
018534
018536
018538
018540
018542
018544
018546
018548
018550
018552
018554
018556
018558
018560
018562
018564
018566
018568
018570
018572
018574
018576
018578
018580
018582
018584
018586
018588
018590
018592
018594
018596
018598
018600
018602
018604
018606
018608
018610
018612
018614
018616
018618
018620
018622
018624
018626
018628
018630
018632
018634
018636
018638
018640
018642
018644
018646
018648
018650
018652
018654
018656
018658
018660
018662
018664
018666
018668
018670
018672
018674
018676
018678
018680
018682
018684
018686
018688
018690
018692
018694
018696
018698
018700
018702
018704
018706
018708
018710
018712
018714
018716
018718
018720
018722
018724
018726
018728
018730
018732
018734
018736
018738
018740
018742
018744
018746
018748
018750
018752
018754
018756
018758
018760
018762
018764
018766
018768
018770
018772
018774
018776
018778
018780
018782
018784
018786
018788
018790
018792
018794
018796
018798
018800
018802
018804
018806
018808
018810
018812
018814
018816
018818
018820
018822
018824
018826
018828
018830
018832
018834
018836
018838
018840
018842
018844
018846
018848
018850
018852
018854
018856
018858
018860
018862
018864
018866
018868
018870
018872
018874
018876
018878
018880
018882
018884
018886
018888
018890
018892
018894
018896
018898
018900
018902
018904
018906
018908
018910
018912
018914
018916
018918
018920
018922
018924
018926
018928
018930
018932
018934
018936
018938
018940
018942
018944
018946
018948
018950
018952
018954
018956
018958
018960
018962
018964
018966
018968
018970
018972
018974
018976
018978
018980
018982
018984
018986
018988
018990
018992
018994
018996
018998
019000
019002
019004
019006
019008
019010
019012
019014
019016
019018
019020
019022
019024
019026
019028
019030
019032
019034
019036
019038
019040
019042
019044
019046
019048
019050
019052
019054
019056
019058
019060
019062
019064
019066
019068
019070
019072
019074
019076
019078
019080
019082
019084
019086
019088
019090
019092
019094
019096
019098
019100
019102
019104
019106
019108
019110
019112
019114
019116
019118
019120
019122
019124
019126
019128
019130
019132
019134
019136
019138
019140
019142
019144
019146
019148
019150
019152
019154
019156
019158
019160
019162
019164
019166
019168
019170
019172
019174
019176
019178
019180
019182
019184
019186
019188
019190
019192
019194
019196
019198
019200
019202
019204
019206
019208
019210
019212
019214
019216
019218
019220
019222
019224
019226
019228
019230
019232
019234
019236
019238
019240
019242
019244
019246
019248
019250
019252
019254
019256
019258
019260
019262
019264
019266
019268
019270
019272
019274
019276
019278
019280
019282
019284
019286
019288
019290
019292
019294
019296
019298
019300
019302
019304
019306
019308
019310
019312
019314
019316
019318
019320
019322
019324
019326
019328
019330
019332
019334
019336
019338
019340
019342
019344
019346
019348
019350
019352
019354
019356
019358
019360
019362
019364
019366
019368
019370
019372
019374
019376
019378
019380
019382
019384
019386
019388
019390
019392
019394
019396
019398
019400
019402
019404
019406
019408
019410
019412
019414
019416
019418
019420
019422
019424
019426
019428
019430
019432
019434
019436
019438
019440
019442
019444
019446
019448
019450
019452
019454
019456
019458
019460
019462
019464
019466
019468
019470
019472
019474
019476
019478
019480
019482
019484
019486
019488
019490
019492
019494
019496
019498
019500
019502
019504
019506
019508
019510
019512
019514
019516
019518
019520
019522
019524
019526
019528
019530
019532
019534
019536
019538
019540
019542
019544
019546
019548
019550
019552
019554
019556
019558
019560
019562
019564
019566
019568
019570
019572
019574
019576
019578
019580
019582
019584
019586
019588
019590
019592
019594
019596
019598
019600
019602
019604
019606
019608
019610
019612
019614
019616
019618
019620
019622
019624
019626
019628
019630
019632
019634
019636
019638
019640
019642
019644
019646
019648
019650
019652
019654
019656
019658
019660
019662
019664
019666
019668
019670
019672
019674
019676
019678
019680
019682
019684
019686
019688
019690
019692
019694
019696
019698
019700
019702
019704
019706
019708
019710
019712
019714
019716
019718
019720
019722
019724
019726
019728
019730
019732
019734
019736
019738
019740
019742
019744
019746
019748
019750
019752
019754
019756
019758
019760
019762
019764
019766
019768
019770
019772
019774
019776
019778
019780
019782
019784
019786
019788
019790
019792
019794
019796
019798
019800
019802
019804
019806
019808
019810
019812
019814
019816
019818
019820
019822
019824
019826
019828
019830
019832
019834
019836
019838
019840
019842
019844
019846
019848
019850
019852
019854
019856
019858
019860
019862
019864
019866
019868
019870
019872
019874
019876
019878
019880
019882
019884
019886
019888
019890
019892
019894
019896
019898
019900
019902
019904
019906
019908
019910
019912
019914
019916
019918
019920
019922
019924
019926
019928
019930
019932
019934
019936
019938
019940
019942
019944
019946
019948
019950
019952
019954
019956
019958
019960
019962
019964
019966
019968
019970
019972
019974
019976
019978
019980
019982
019984
019986
019988
019990
019992
019994
019996
019998
020000

```



```

3  CONTINUE
   PRINT, "FREQUENCY RESPONSE FOR BALANCED SYSTEM"
   PRINT, " "
   PRINT, "THE BALANCED A MATRIX MUST BE CREATED BY OPTION"
   PRINT, "2 BEFORE THIS OPTION CAN BE EXECUTED!!"
   PRINT, "THIS OPTION PLOTS THE FREQUENCY RESPONSE TO A"
   PRINT, "GIVEN UNDER SYSTEM. ENTER DESIRED ORDER > "
   READ, IS-T
   IF (IS-T.L.E.0.0K.1511.51.4) PRINT, "ILLEGAL ORDER"
   IF (IS-T.L.C.0.0K.1511.51.4) GO TO 2
   PRINT, "SYSTEM HAS ", IS-T, " INPUTS AND ", MC, " OUTPUTS."
   PRINT, "SELECT DESIRED INPUT-OUTPUT COMBINATION > "
   READ, I-L-K
   PRINT, "PLEASE ENTER 4M14, -3 FOR .C11, -2 FOR .C1 ETC > "
   READ, I-M-T
   M1-M=1.0E-11
   DO 97 J=1, 1'
   DO 97 JO=1, 1'
   CREAL(IJ, JO) = AREAL(IJ, JO)
97  CC(I) INDE
   CALL L1GKF(CREAL, ISTT, 1J, 1, EIG, EIGV, IC, WK, IER)
   DO 97 I=1, N
   DO 97 J=1, N
   REIGV(I, J) = (0.0E0, 0.0E0)
   REIGV(I, I) = (1.0, 0.0)
   REIGV(I, J) = E*GV(I, J)
   CC(I) INDE
5  CALL L5OTIC(00, ISTT, 1J, REIGV, ISTT, 10, L, WK, IER)
   DO 44 I=1, 102
   MAGG(M) = 0.0E0
   ARGG(M) = 0.0E0
   DO 44 J=1, 13'
   GG(M) = (1.0E0, 0.0E0)
   GC(M) = (1.0E0, 0.0E0)
   CC(I) INDE
4  DO 44 J=1, 1511
   SUM15 = (0.15E0, 0.0E0)
   SUM15 = (0.15E0, 0.0E0)
   SUM15 = SUM15 + (REIGV(J, I), REIGV(I, L))

```

```

010100
010110
010120
010130
010140
010150
010160
010170
010180
010190
010200
010210
010220
010230
010240
010250
010260
010270
010280
010290
010300
010310
010320
010330
010340
010350
010360
010370
010380
010390
010400
010410
010420
010430
010440
010450
010460
010470
010480
010490

```

```

90 SU4C=SU4C+(CDAR(K,I)*ZIGV(I,J))
   CONTINUE
   K3(J)=3*HC*SU4C
   WINDIC=1
   WMIN=WMIN
   W(1,WINDIC)=WMIN
   ZC(WINDIC)=ZC(J)-(CMPLX(1.0,0.0)/(CMPLX(0.0,W(WINDIC))-ZIG(J)))
   G0(WINDIC)=G0(WINDIC)+RC(WINDIC)
   IF(WINDIC.GE.100)GO TO 131
   W(41,WINDIC+1)=W(WINDIC)+WMIN
   WINDIC=WINDIC+1
   GO TO 92
CONTINUE
DO 92K1=1,WINDIC
CALL TOLAM(MAG,ANG,G6(K1))
MAG=2.0*LOG10(MAG)
MAGG(K1)=MAG
ANGG(K1)=ANG
92 CONTINUE
PRINT 94,"K," INPUT-OUTPUT COMBINATION TRANSFER FUNCTION"
PRINT,"FREQUENCY RESPONSE PLOT FOR ",IST1," ORDER SYSTEM"
PRINT,"MAGNITUDE PLOT (DB)"
CALL TTLES(ID1(13))
PRINT,"SPACE PLOT (DB)"
CALL TTLES(ID2(13))
CALL PLOT5(CAT,NUM,MJM)
CALL PLOT(CAT,NUM,-3)
CALL PLOT(CAT,NUM,3,-3)
CALL MAGRAPH1(MAGGG,WINDIC,ID1,10,0)
CALL MAGRAPH1(MAGGG,WINDIC,ID2,10,0)
CALL PLOT5(WINDIC)
WRITE(WINDIC/2
PRINT,"DO YOU WISH TO SUPPRESS THE LISTING(1=YES,2=NO) > "
READ,IOST
IF(1005.70.5) SUP=.TRUE.
IF(SUP)GO TO 9999
WRITE(6,34)
FORMAT(1,14,"",123,"MAG(DB)",139,"ANGLE(DEG)")
DO 94J=1,WINDIC
WRITE(6,30) W(K1),MAGG(K1),ANGG(K1)
94

```

```

010500
010510
010520
010530
010540
010550
010560
010570
010580
010590
010600
010610
010620
010630
010640
010650
010660
010670
010680
010690
010700
010710
010720
010730
010740
010750
010760
010770
010780
010790
010800
010810

```

```

96   FORMAT(IX,TI,67.2,T20,E13.4,T35,E13.4)
93   CONTINUE
599  CONTINUE
C-----
C   THIS SUBROUTINE FINDS THE MAG AND ANGLE OF A COMPLEX NUMBER
C-----
SUBROUTINE POLAR(MAG,ANG,KATELE)
COMPLEX X,NATLE
REAL MAG,IMAG
RHAG=REAL(NATLE)
IMAG=AIMAG(NATLE)
MAG=SQRT(RHAG**2+IMAG**2)
IF(RHAG) 1,5
IF(1.156) 2,3,4
1   ANG=-1.
2   ANG=-1.
3   RETURN
4   ANG=-27.
5   RETURN
6   ANG=ATAN(IMAG/RHAG)*(180./3.1415926535)
7   RETURN
8   ANG=ATAN(IMAG/RHAG)*(180./3.1415926535)-160.
9   RETURN
10  END
C-----
C   THIS SUBROUTINE PLOTS ROSE PLOTS
C-----
SUBROUTINE HGRAPH(X,Y,N,IO,NU,NP,NS)
DIMENSION X(1),Y(1),ID(1)
IF(NO.EQ.2) GO TO 30
IF(NU.LT.3) GO TO 10
CALL SCALE(X,Y,N,1)
CALL PLOT(5.5,5.5,3)
CALL PLOT(-1.35,1.35,3)
CALL PLOT(-.15,1.55,2)
IF(17(1).EQ.0) GO TO 25
CALL PLOT(-.05,5.55,3)
DO 2 IF=1,7,2
CALL SYH30L(I*1-0.9,1.85,.07,IO(I),J(.,20)

```

```

018900
018910
018920
018930
018940
018950
018960
018970
018980
018990
019000
019010
019020
019030
019040
019050
019060
019070
019080
019090
019100
019110
019120
019130
019140
019150
019160
019170
019180
019190
019200
019210
019220
019230
019240
019250
019260
019270
019280
019290

```


[illegible]


```

021700 PRINT, "THE CGAR MATRIX IS "
021710 CALL PRINTM(MC,NC,CBAR)
021720 PRINT, "PLEASE ENTER THE SAMPLING TIME > "
021730 READ, TSAMP
021740 CALL DKRECT(F,G,N,M3,M3,MKAREA,ABAR,99AR,TSAMP)
021750 PRINT, "THE F MATRIX IS "
021760 CALL PRINTM(M,N,F)
021770 PRINT, "THE G MATRIX IS "
021780 CALL PRINTM(MB,MB,G)
021790 PRINT, "DO YOU WISH TO OBTAIN THE DISCRETE U3S. MAT(1=YES,2=NO) > "
021800 READ, INVIC
021810 IF(1)GOTO 10,2)GOTO 1399
021820 CALL U3S(MC,NC,N,F,MINVED,MORAREA,FMOR,CBAR,M2)
021830 PRINT, "THE OBS. MAT. IS "
021840 CALL PRINTM(M2,N,M,M03)
021850 CONTINUE
021860 EN)
021870 C-----
021880 C THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
021890 C-----
021900 SUBROUTINE PRINTM(M,N,M03)
021910 DIMENSION DPAT(10,10)
021920 DO 201=1,M,1
021930 WRITE (5,10) I,(DPAT(I,J),J=1,N,1)
021940 FORMAT(145,12,(T4,4(1P13.4,1X)))
021950 CONTINUE
021960 RETURN
021970 EN)
021980 C-----
021990 C THIS SUBROUTINE MULTIPLIES TWO MATRICES AND STORES IN C
022000 C-----
022010 SUBROUTINE PMULT(A,B,22L,M,N)
022020 DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
022030 DO 21=1,L
022040 DO 22=1,M
022050 SUM4=0.0
022060 DO 10=1,N
022070 SUM4=SUM4+A(I,J)*B(J,K)
022080 D(I,K)=SUM4
022090 DO 31=1,L

```

```

3      DO 3 J=1,N
      C(I,J)=D(I,J)
      RETURN
      END
C-----
C      THIS SUBROUTINE PRINTS OUT THE CONT AND OBS MATRICES
C-----
      SUBROUTINE PRINTM1(M,4,EMAT)
      DIMENSION EMAT(30,10)
      DO 20 I=1,M,2
      WRITE (5,10) I, (EMAT(I,J), J=1,4,1)
10      FORMAT(1H0,72,(T4,4(I)E13.4,1X))
20      CONTINUE
      RETURN
      END
C-----
C      THIS SUBROUTINE FINDS EXP(AMAT*1) AND ITS INTEGRAL
C      VIA A TRUNCATED INFINITE SERIES APPROXIMATION
C-----
      SUBROUTINE DISMPEI(M,A,DEL,EA,EAINI,NI)
      DIMENSION EA(10,10),EAINI(10,10),A(10,10),Z(10,10),
      CHINMED(1,1),CGR(1,10)
      DO 10 I=1,M
      DO 11 J=1,M
      EAINI(I,J)=*.00P
      EAINI(I,I)=DEL
      Z(I,J)=EAINI(I,J)
10      CONTINUE
      DO 20 K=2,NT
      DO 13 I=1,M
      DO 13 J=1,M
      GRAR(I,J)=A(I,J)* (DEL/FLCAT(K))
20      CONTINUE
      CALL MPUT(7,CGR,MINMED,M,N,M)
      DO 16 I=1,M
      DO 16 J=1,M
      Z(I,J)=Z(I,J)+GRAR(I,J)
      EAINI(I,J)=EAINI(I,J)+MINMED(I,J)
20      CONTINUE
400      CONTINUE

```

```

022100
022110
022120
022130
022140
022150
022160
022170
022180
022190
022200
022210
022220
022230
022240
022250
022260
022270
022280
022290
022300
022310
022320
022330
022340
022350
022360
022370
022380
022390
022400
022410
022420
022430
022440
022450
022460
022470
022480
022490

```

```

022580 CALL MHULT(P,EAINT,Z,I,N,N)
022581 DO 501 I=1,N
022582 DO 501 J=1,N
022583 EA(I,J)=0.0
022584 EA(I,I)=1.0
022585 CONTINUE
022586 DO 601 I=1,N
022587 DO 601 J=1,N
022588 EA(I,J)=EA(I,J)+Z(I,J)
022589 CONTINUE
022590 RETURN
022591 END

022600 C-----
022601 C THIS SUBROUTINE ACCEPTS EXP(AHAT*T) AND ITS INTEGRAL
022602 C AND FORMS THE DISCRETE F MATRIX AND G MATRIX
022603 C-----
022604 SUBROUTINE MKREFT(F,G,M,N,IB,NS,MKAREA,A,B,T)
022605 DIMENSION F(10,10),G(10,10),MKAREA(10,10),A(10,10)
022606 DIMENSION R(10,10)
022607 CALL DISKRE(M,A,T,F,MKAREA,S(I)
022608 CALL MHULT(MKAREA,G,G,M,N,IB,NS)
022609 RETURN
022610 END

022620 C-----
022621 C THIS SUBROUTINE FINDS THE DISCRETE CONTROLLABILITY
022622 C MATRIX
022623 C-----
022624 SUBROUTINE COMBLY(F,G,M3,NB,N,3-DD,MOPAREA,WINMED,M1)
022625 DIMENSION F(10,10),G(10,10),P4DD(10,30),WINMED(10,10)
022626 DIMENSION MKAREA(10,10)
022627 DO 101 I=1,NB
022628 DO 101 J=1,NB
022629 P4DD(I,J)=G(I,J)
022630 CONTINUE
022631 NB=N3
022632 IF(NB-50.1) GO TO 150
022633 CALL MHULT(F,G,WINMED,M,N,NB)
022634 DO 12 I=1,NB
022635 DO 12 J=1,NB
022636 K=J

```

```

022900      M1=M2
022910      WOOD(I,M2+J)=WINMED(I,J)
022920      CONTINUE
022930      M1=M1+K
022940      IF(M1-2)GO TO 150
022950      M1=M1-1
022960      DO 11 L=2,M1,1
022970      CALL MULT(F,WINMED,W)RAREA,N,M,NB)
022980      DO 13 I=1,M1,1
022990      DO 13 J=1,M1,1
023000      WOOD(I,M1+J)=WOKAREA(I,J)
023010      WINMED(I,J)=WOKAREA(I,J)
023020      K=J
023030      CONTINUE
023040      M1=M1+K
023050      CONTINUE
023060      REFUG=
023070      ENJ
023080      C-----
023090      C THIS SUBROUTINE FINDS THE DISCRETE OBSERVABILITY MATRIX
023100      C-----
023110      SUBROUTINE OBS(MC,NC,N,F,WINMED,WOKAREA,RPOB,C,M2)
023120      DIMENSION WINMED(11,11),WOKAREA(11,10),C(11,10)
023130      DIMENSION F(11,10),WPOB(11,10)
023140      DO 21 I=1,NC
023150      DO 21 J=1,NC
023160      WPOB(I,J)=C(I,J)
023170      M2=MC
023180      CONTINUE
023190      IF(M1-1)GO TO 250
023200      CALL MULT(C,F,WINMED,WOKAREA,NC,NC,N)
023210      DO 22 I=1,NC
023220      DO 22 J=1,NC
023230      K=I
023240      M2=MC
023250      WPOB(N2+1,J)=WINMED(I,J)
023260      CONTINUE
023270      M2=M2+K
023280      IF(M1-2)GO TO 250
023290      M1=M1-1

```



```

023700 CALL ZMINV(P2,M,KMCC,P,N2,0)
023710 CALL VMULFF(RMUT,F,N2,N2,10,10,WINMED,10,IER)
023720 CALL VMULFF(WINMED,P,N2,N2,10,10,ABAR,10,IER)
023730 CALL VMULFF(RMUT,G,N2,N2,10,10,ABAR,10,IER)
023740 CALL VMULFF(C,P,N2,N2,10,10,CBAR,10,IER)
023750 PRINT,"THE F MATRIX IS "
023760 CALL PRINTM(N2,N2,ABAR)
023770 PRINT,"THE G MATRIX IS "
023780 CALL PRINTM(N2,N2,ABAR)
023790 PRINT,"THE C MATRIX IS "
023800 CALL PRINTM(N2,N2,CBAR)
023810 CONTINUE
023820 ENJ
023830 C-----
023840 C THIS SUBROUTINE PRINTS OUT A MATRIX FOR VERIFICATION
023850 C-----
023860 SUBROUTINE PRINTM(M,N,DMAT)
023870 DIMENSION OPAT(10,10)
023880 DO 20 I=1,M,1
023890 WRITE (5,10) I,(DMAT(I,J),J=1,N,1)
023900 FORMAT(10,10,12,('4,4,10F13.4,1X'))
023910 CONTINUE
023920 RETURN
023930 ENJ
023940 C-----
023950 C THIS SUBROUTINE MULTIPLIES TWO MATRICES AND STORES IN C
023960 C-----
023970 SUBROUTINE PMULT(A,B,C,L,M,N)
023980 DIMENSION A(10,10),B(10,10),C(10,10),D(10,10)
023990 DO 20 I=1,L
024000 DO 20 K=1,M
024010 SUM=0.0
024020 DO 30 J=1,N
024030 SUM=SUM+A(I,J)*B(J,K)
024040 D(I,K)=SUM
024050 DO 30 J=1,N
024060 C(I,J)=D(I,J)
024070 RETURN
024080 ENJ
024090

```

D-61

```

500      EA(I,J)=1.0/0
      CONTINUE
      DO 501 I=1,N
      DO 502 J=1,N
      EA(I,J)=EA(I,J)+Z(I,J)
500      CONTINUE
      RETURN
      END

C-----
C      THIS SUBROUTINE ACCEPTS EXP(MAT*Y) AND ITS INTEGRAL
C      AND FORMS THE DISCRETE F MATRIX AND G MATRIX
C-----
C      SUBROUTINE CKRLET(F,G,N,M,NB,NB,MCO,MOKAKEA,A,B,I)
C      DIMENSION F(10,10),G(10,10),MOKAKEA(10,10),A(10,10)
C      DIMENSION B(10,10)
C      CALL CKRLET(N,M,NB,NB,MCO,MOKAKEA,A,B,I)
C      CALL MULT(MOKAKEA,B,G,N,NB,NB)
C      RETURN
      END

C-----
C      THIS SUBROUTINE FINDS THE DISCRETE CONTROLLABILITY
C      MATRIX
C-----
C      SUBROUTINE CONTY(F,G,M,NB,NB,MCO,MOKAKEA,WINMED,N1)
C      DIMENSION F(10,10),G(10,10),MCO(10,10),WINMED(10,10)
C      DIMENSION MOKAKEA(10,10)
C      DO 11 I=1,M
C      DO 11 J=1,N1
C      MCO(I,J)=G(I,J)
100      CONTINUE
      M1=N1
      IF(N1(0.1)GG TO 101
      CALL MULT(F,G,WINMED,N,N1)
      DO 12 J=1,N1
      DO 12 I=1,N1
      K=J
      M1=M1
      K=J(1,N1+J)=WINMED(I,J)
120      CONTINUE
      M1=M1+K

```

```

139 IF(N,LO,2)GO TO 150
140 AN=N-1
141 DO 14 L=2,NR,1
142 CALL FMULT(F,WINNED,MORAREA,N,N,NR)
143 DO 13 I=1,M,1
144 DO 13 J=1,N,1
145 RMJ(I,N1+J)=MORAREA(I,J)
146 WINNED(I,J)=MORAREA(I,J)
147 K=J
148 CONTINUE
149 N1=N1+K
150 CONTINUE
151 RETURN
152 ENJ
153 C.....
154 C THIS SUBROUTINE FINDS THE DISCRETE OBSERVABILITY MATRIX
155 C.....
156 SUBROUTINE COS(MC,NC,I,F,WINNED,MORAREA,RM00,C,N2)
157 DIMENSION WINNED(10,10),MORAREA(10,10),C(10,10)
158 DIMENSION F(10,10),K0B(10,10)
159 DO 21 I=1,MC
160 DO 21 J=1,NC
161 KM3(I,J)=C(I,J)
162 N2=MC
163 CONTINUE
164 IF(N,LO,1)GO TO 250
165 CALL FMULT(C,F,WINNED,MC,NC,N)
166 DO 21 I=1,MC
167 DO 22 J=1,NC
168 K=I
169 N2=MC
170 KM3(I,2+I,J)=WINNED(I,J)
171 CONTINUE
172 N2=N2+K
173 IF(N,LO,2)GO TO 250
174 AN=N-1
175 DO 21 L=2,NR,1
176 CALL FMULT(WINNED,F,MORAREA,MC,NC,N)
177 DO 21 I=1,MC
178 DO 23 J=1,NC

```

```

      RHOD(N2+I,J)=WOFAREA(I,J)
      WJMED(I,J)=WOFAREA(I,J)
      K=I
232  CONTINUE
      N2=N2+K
240  CONTINUE
250  KSTU2?
      EN?
C*****
      OVERLAY(17,0)
C*****
      PRJGR=3 GRAPH
      COMMON/HAZIN/NUIM,NOI41,COM1/INOU/KIN,KOUT,KPUNCH
      CCHN3/ATMO1/A(10,10),M,N,B(10,10),M,N,N,F(1,10),G(10,10),C(10,10),WSQUAR(10,10)
      *,10,NC
      CCHN3/HENOT1/ATRAN(10,10),BTAN(10,10),CIRAN(10,10),WSQUAR(10,10)
      *,M,OUT(2(10,10),BPROT(10,10),CPAOD(10,10)
      DO 6 I=1,4
      DO 6 J=1,4
      ATRAN(J,I)=A(I,J)
      CONTINUE
      DO 7 I=1,M
      DO 7 J=1,N
      BTAN(J,I)=B(I,J)
      CONTINUE
      DO 8 I=1,AC
      DO 8 J=1,NC
      CIRAN(J,I)=C(I,J)
      CONTINUE
      CALL WHILFF(B,BTRAN,M3,N3,M3,10,10,BPROD,10,IER)
      CALL WHILFF(CTRAN,CNC,MC,NC,10,10,CPROD,10,IER)
      CALL MLJECIN,ATRAN,BPOD,WSQUAR,.01)
      CALL MLJECIN(MC,A,CPOD,WSQUAR2,.01)
      PRINT," "
      PRINT,"THE CONTROLLABILITY GRAPHIAN IS "
      CALL PLJTH(M,N,WSQUAR)
      PRINT," "
      PRINT,"THE OBSERVABILITY GRAPHIAN IS "
      CALL PLJTH(M,N,WSQUAR2)
      EN?

```

025700
025710
025720
025730
025740
025750
025760
025770
025780
025790
025800

```

C.....
C  THIS PROGRAM PRINTS OUT A GIVEN MATRIX
C.....
      SUBROUTINE PRINTM(M,N,OMAT)
      DIMENSION OMAT(10,10)
      DO 2 J=1,M+1
      WRITE (5,10) I,OMAT(I,J),J=1,N+1
      FORMAT(1H0,12,(T4,413F13.4,1X))
      CONTINUE
      RETURN
      END
16
26

```

VITA

James Robert McClendon was born on 12 October 1955 in Aruba, Netherlands Antilles. He was graduated from Spartanburg High School in 1973 and attended Clemson University from which he received a Bachelor of Science in Electrical and Computer Engineering with honors in May 1978. Upon graduation, he was designated a distinguished graduate and received a commission in the USAF through the AFROTC program. He entered active duty in June 1978 and was assigned to the School of Engineering, Air Force Institute of Technology.

Permanent Address: 155 Marlin Drive
Spartanburg, South Carolina
29302

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/79-22	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MODEL ORDER REDUCTION USING THE BALANCED REPRESENTATION: THEORY, APPLICATION, AND INTERACTIVE SOFTWARE IMPLEMENTATION		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
7. AUTHOR(s) James R. McClendon 2nd Lt, USAF		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Flight Dynamics Laboratory Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1979
		13. NUMBER OF PAGES 211
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release; IAW AFR 190-17 Joseph P. Hipps, Major, USAF Director of Information		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) SINGULAR VALUE ANALYSIS MODEL ORDER REDUCTION SYSTEM MODELLING INTERACTIVE COMPUTER PROGRAM GRAMMIANS MATHEMATICAL MODELLING		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report investigates a model order reduction technique developed by Dr Bruce Moore of the University of Toronto, as applied to the B-52E flutter control problem currently under study by the Air Force Flight Dynamics Laboratory. The algorithm, which is based upon singular value analysis, is applied to the full twenty-fourth order model yielding an internally balanced representation which is balanced with respect to controllability and observability properties. The		

DD FORM 1473
1 JAN 73

EDITION OF 1 NOV 68 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

✓ system is reduced in order, and comparisons are made between the Moore algorithm model and that obtained via a method used by the Air Force Flight Dynamics Laboratory.

In addition to this investigation, an interactive program is presented which contains the model order reduction algorithm. Other capabilities include: estimation of the condition number with respect to inversion, singular value and condition number plotting vs. sample time for discrete time controllability, observability, and Hankel matrices, frequency response generation, and various special coordinate system transformations. ↗

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)